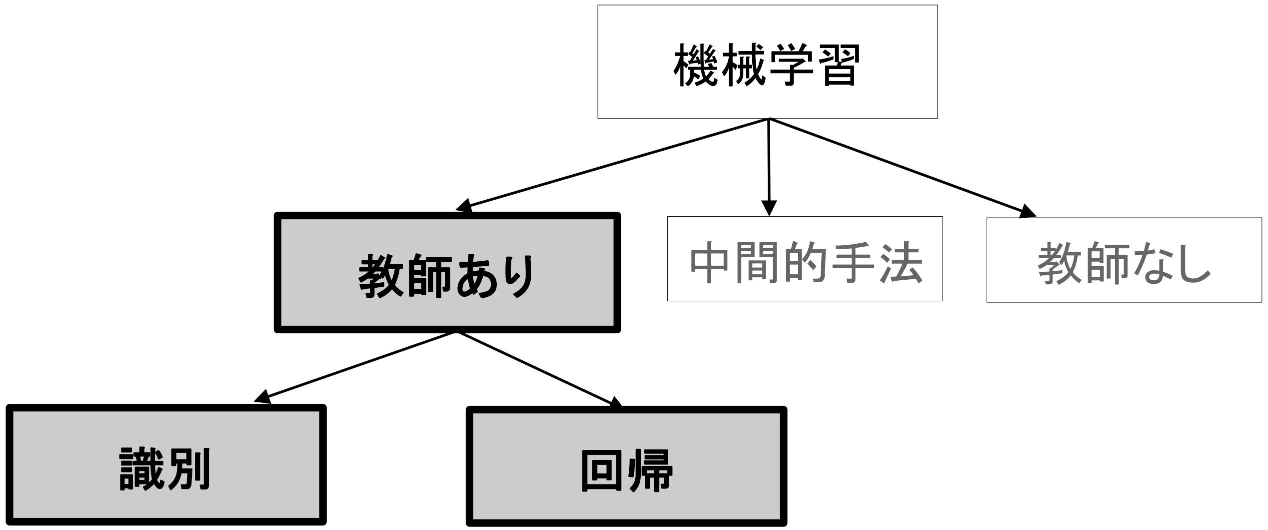
第7章～第10章 教師あり学習の発展的手法

- 識別と回帰のいずれにも適用可能
- 線型モデルでは高い性能が得られないデータに適する



- サポートベクトルマシン
- ニューラルネットワーク（深層学習）
- アンサンブル学習

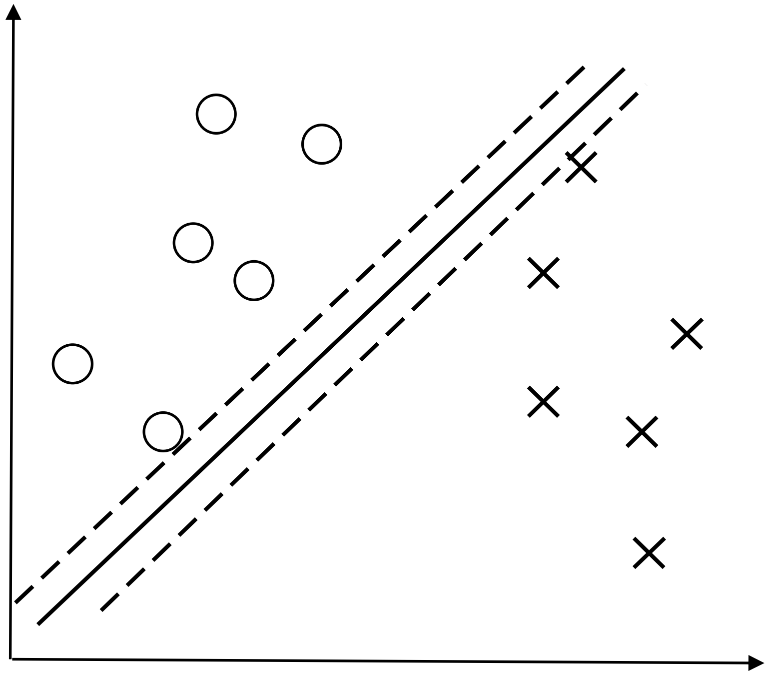
7. サポートベクトルマシン

- 本章の説明手順

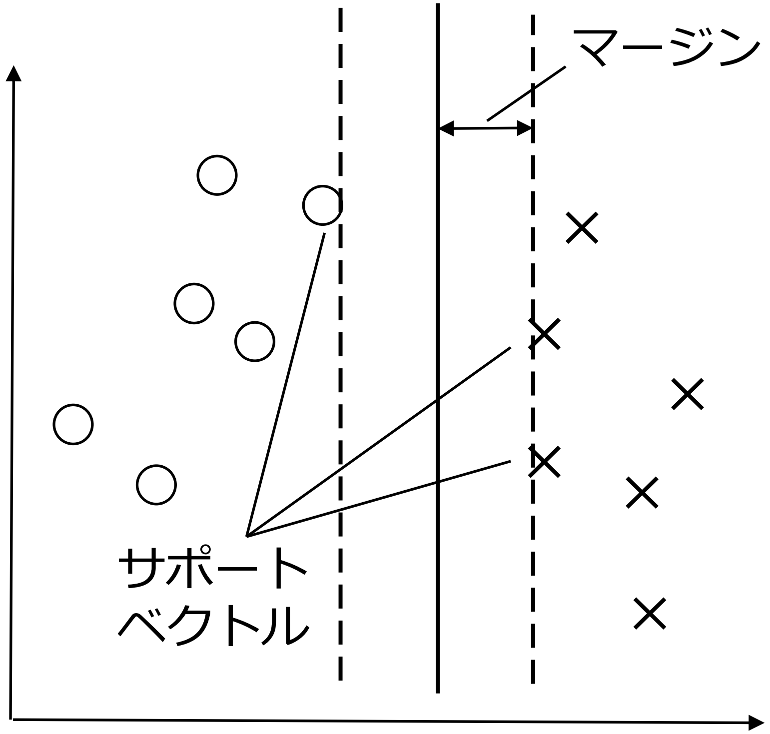
1. 線形分離可能なデータに対して，なるべく学習データに特化しすぎない識別面を求める
2. 線形分離不可能なデータに対して，誤識別に対するペナルティを設定することで、1. の手法を改良する
3. さらに複雑なデータに対して、学習データを高次元の空間に写して，2. の手法を適用する
4. 3. の手法に対して、最適なハイパーパラメータを求める

7.1 サポートベクトルマシンとは

- 汎化性能を高めるために、マージンを最大化する識別面を求める
  - ◆ マージン：識別面と、もっとも近いデータとの距離
  - ◆ 学習データは線形分離可能とする



(a)マージンの小さい識別面



(b)マージンの大きい識別面

7.1.1 マージン最大化のための定式化

- 学習データ

$$\{(\boldsymbol{x}_i, y_i)\} \quad i = 1, \ldots, N, \quad y_i = 1 \text{ or } -1$$

- 識別面の式（超平面）

$$\boldsymbol{w}^T \boldsymbol{x}_i + w_0 = 0$$

w, x はd次元

- 識別面の制約を導入（係数を定数倍しても超平面は不変）

$$\min_{i=1,\ldots,N} |\boldsymbol{w}^T \boldsymbol{x}_i + w_0| = 1$$

- 学習データと識別面との最小距離（マージン）

これを最大化

$$\min_{i=1,\ldots,N} \mathrm{Dist}(\boldsymbol{x}_i) = \min_{i=1,\ldots,N} \frac{|\boldsymbol{w}^T \boldsymbol{x}_i + w_0|}{||\boldsymbol{w}||} = \frac{1}{||\boldsymbol{w}||}$$

点と直線の距離の公式

$$r = \frac{|ax + by + c|}{\sqrt{a^2 + b^2}}$$

7.1.1 マージン最大化のための定式化

- 目的関数： $\min \frac{1}{2}||\boldsymbol{w}||^2$

1 / $||\boldsymbol{w}||$ の最大化を $||\boldsymbol{w}||^2$ の最小化と置き換え

- 制約条件： $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) \geq 1 \quad i = 1, \ldots, N$
- 解法：ラグランジュの未定乗数法
  - ◆ 例題（2変数、等式制約） $\min f(x,y) \quad s.t. \; g(x,y) = 0$
  - ◆ ラグランジュ関数 $L(x,y,\alpha) = f(x,y) + \alpha g(x,y)$
    - ラグランジュ係数の制約 $\alpha \geqq 0$
    - L を x, y, α で偏微分して0になる値が f の極値

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} = \frac{\partial L}{\partial \alpha} = 0$$

変数3つに対して制約式3つ

7.1.2 マージンを最大とする識別面の計算

- より解きやすい問題への変換

$$L(\boldsymbol{w}, w_0, \boldsymbol{\alpha}) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{i=1}^{N} \alpha_i (y_i (\boldsymbol{w}^T \boldsymbol{x} + w_0) - 1)$$

$$\frac{\partial L}{\partial w_0} = 0 \quad \Rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \quad \Rightarrow \boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i$$

$$L(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j - \sum_{i=1}^{N} \alpha_i$$

この式の最小化は $\boldsymbol{\alpha}$ についての2次計画問題なので極値をとる $\boldsymbol{\alpha}$ が求まる

7.1.2 マージンを最大とする識別面の計算

- 定数項の計算
  - 各クラスのサポートベクトルから求める

$$w_0 = -\frac{1}{2}(\boldsymbol{w}^T\boldsymbol{x}_+ + \boldsymbol{w}^T\boldsymbol{x}_-)$$

- マージンが最大の識別関数

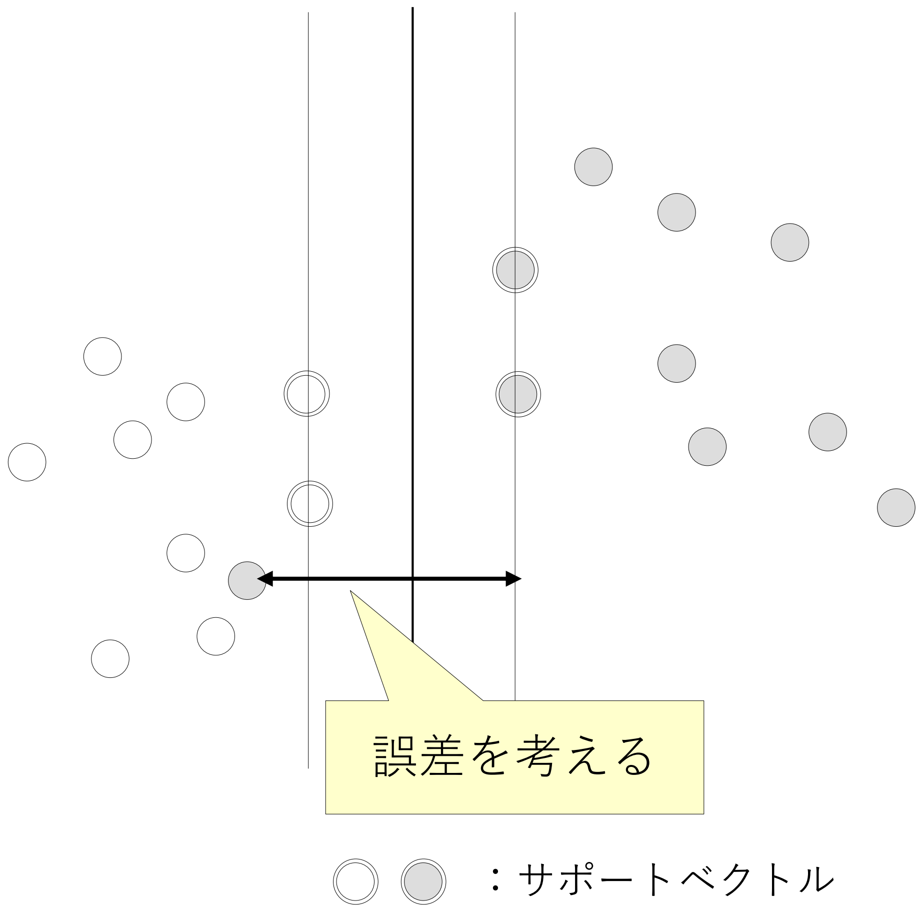
$$g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$$
$$= \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}^T\boldsymbol{x}_i + w_0$$

サポートベクトルに対応する a_i のみが 0 以上、残りは 0

マージン最大の識別面の決定には
サポートベクトルしか関与しない

7.2 ソフトマージンによる誤識別データの吸収

- 少量のデータが線形分離性を妨げている場合



7.2 ソフトマージンによる誤識別データの吸収

- スラック変数 ξ_i の導入

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) \geq 1 - \xi_i \quad i = 1, \ldots, N$$

ξの値
0: マージンの外側
0<ξ≤1:マージンの内側
1<ξ : 誤り

- 最小化問題の修正

$$\min(\frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{N}\xi_i)$$

スラック変数も
小さい方がよい

- 計算結果
  - α_iの2次計画問題に $0 \leq \alpha_i \leq C$ が加わるだけ

7.2 ソフトマージンによる誤識別データの吸収

- C: エラー事例に対するペナルティ
  - ◆ 大きな値：誤識別データの影響が大きい

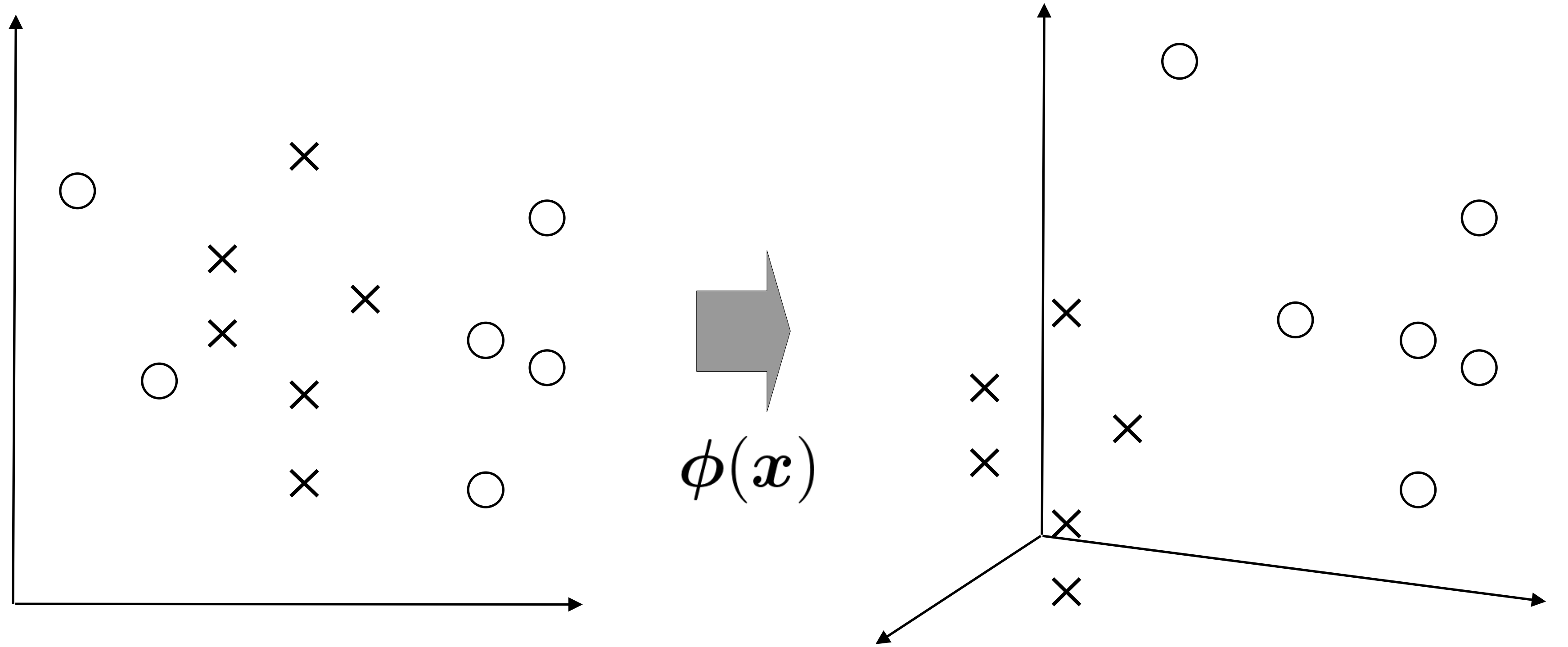
    → 複雑な識別面
  - ◆ 小さな値：誤識別データの影響が小さい

    → 単純な識別面

7.3 カーネル関数を用いたSVM

- クラスが複雑に入り交じった学習データ

  ⇒ 特徴ベクトルを高次元空間に写像



もとの次元で線形分離不可能なデータ

線形分離可能性の高い高次元へ

ただし、もとの空間でのデータ間の近接関係は保持するように

7.3 カーネル関数を用いたSVM

- 非線形変換関数： $\phi(\boldsymbol{x})$
- カーネル関数

$$K(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})^T \phi(\boldsymbol{x}')$$

2つの引数値の近さを表す

  - もとの空間での近さが、変換後の空間の内積に対応
  - $\boldsymbol{x}$ と $\boldsymbol{x}'$ が近ければ $K(\boldsymbol{x}, \boldsymbol{x}')$ は大きい値

7.3 カーネル関数を用いたSVM

- カーネル関数の例（scikit-learnの定義）
  - ◆ 線形　$K(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^T \boldsymbol{x}'$
    - もとの特徴空間でマージン最大の平面
  - ◆ 多項式　$K(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^T \boldsymbol{x}' + r)^d$
    - d 項の相関を加える。r はたいてい1
  - ◆ RBF（ガウシアン）　$K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma ||\boldsymbol{x} - \boldsymbol{x}'||^2)$
    - γ の値：大→複雑　小→単純な識別面
  - ◆ シグモイド　$K(\boldsymbol{x}, \boldsymbol{x}') = \tanh(\boldsymbol{x}^T \boldsymbol{x}' + r)$
    - ベクトルの近さに対して閾値関数的な振る舞い

7.3 カーネル関数を用いたSVM

- 線形カーネルの解釈

$$\boldsymbol{x}^T\boldsymbol{x}' = ||\boldsymbol{x}|| \cdot ||\boldsymbol{x}'|| \cdot \cos\theta$$

θが0に近い（＝ベクトルとして似ている）ほど大きな値

- 多項式カーネルの展開例（$\boldsymbol{x}$ が2次元の場合）

$$\begin{aligned}
K(\boldsymbol{x}, \boldsymbol{x}') &= (\boldsymbol{x}^T\boldsymbol{x}' + 1)^2 \\
&= (x_1x_1' + x_2x_2' + 1)^2 \\
&= (x_1x_1')^2 + (x_2x_2')^2 + 2x_1x_1'x_2x_2' + 2x_1x_1' + 2x_2x_2' + 1 \\
&= ((x_1)^2, (x_2)^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1) \\
&\quad \cdot ((x_1')^2, (x_2')^2, \sqrt{2}x_1'x_2', \sqrt{2}x_1', \sqrt{2}x_2', 1)
\end{aligned}$$

6次元空間に写像されている

7.3 カーネル関数を用いたSVM

- 変換後の線形識別関数：　$g(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) + w_0$
- SVMで求めた $\boldsymbol{w}$ の値を代入

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{\phi}(\boldsymbol{x}_i)$$

$$g(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\phi}(\boldsymbol{x}_i) + w_0$$

$$= \sum_{i=1}^{N} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + w_0$$

非線形変換の式は不要！！！

カーネルトリック

◆ 変換後の空間での線形識別面は、もとの空間での複雑な非線形識別面に対応

7.3 カーネル関数を用いたSVM

- sklearnの学習パラメータ

```
SVC(
    C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape=None, degree=3, gamma='auto',
    kernel='rbf',max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
```

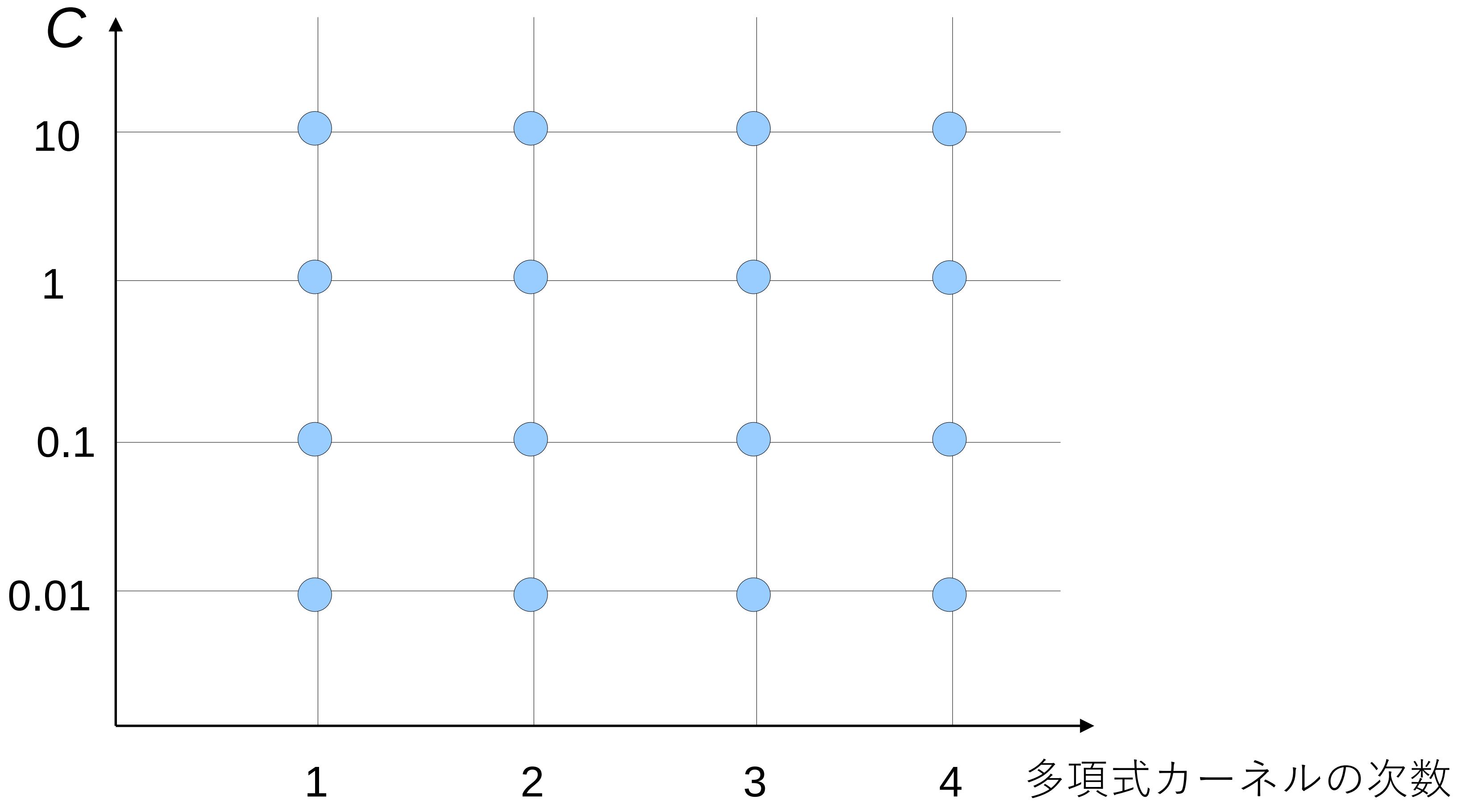
- kernel: カーネル
  - ‘linear’: 線形カーネル
  - ‘poly’: 多項式カーネル
  - ‘rbf’ : RBFカーネル
  - その他: ‘sigmoid’, ‘precomputed’
  - degreeはpolyカーネルを指定したときの次数
  - gamma は（主として）RBFカーネルを指定したときの係数

7.5 ハイパーパラメータのグリッドサーチ

- パラメータ：学習データから学習可能
  - ◆ 識別関数の重み $\boldsymbol{w}$
  - ◆ SVMの $\boldsymbol{a}$
  - ◆ ニューラルネットワークの結合の重み
- ハイパーパラメータ：学習前に決めておく
  - ◆ 識別関数の次数 b、正則化項の重み a
  - ◆ SVM スラックス変数の重み C, 多項式カーネルの次数 d
  - ◆ ニューラルネットワークの中間ユニット数、層数

7.5 ハイパーパラメータのグリッドサーチ

- ハイパーパラメータが複数ある場合
  - グリッドサーチ：各格子点で性能を予測する



サポートベクトル回帰

- 基底関数にカーネルを用いる

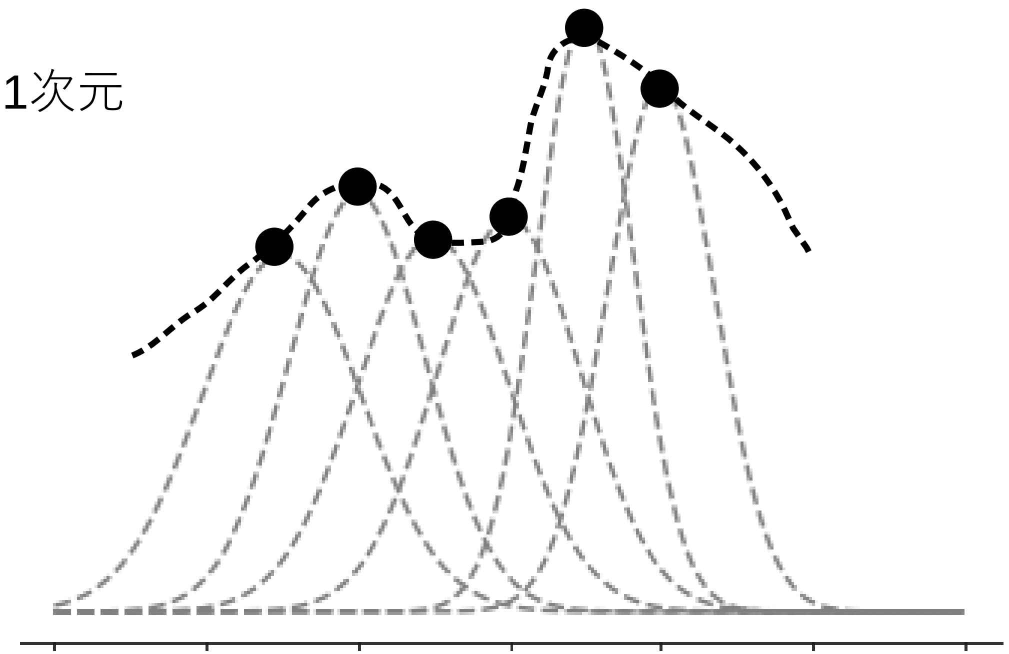
$$\hat{c}(\boldsymbol{x}) = \sum_{j=1}^{N} \alpha_j K(\boldsymbol{x}, \boldsymbol{x}_j)$$

◆ RBFカーネルを用いた場合

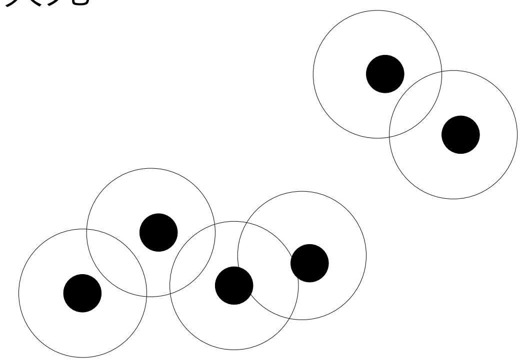
$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma ||\boldsymbol{x} - \boldsymbol{x}'||^2)$$

近くにある学習データとのカーネル関数の値の重み付き和
＝学習データの近傍でのみ関数を近似

1次元



2次元



まとめ

- SVMは線形分離可能なデータに対して，マージン最大の識別面を求める手法
- 誤識別に対するペナルティをスラック変数として設定することで、線形分離不可能なデータへも適用可能
- 学習データを高次元の空間に写像するカーネル法では、カーネルトリックが使える
- SVMのような多数のハイパーパラメータを持つモデルではグリッドサーチを行う

補足

ラグランジュの未定乗数法

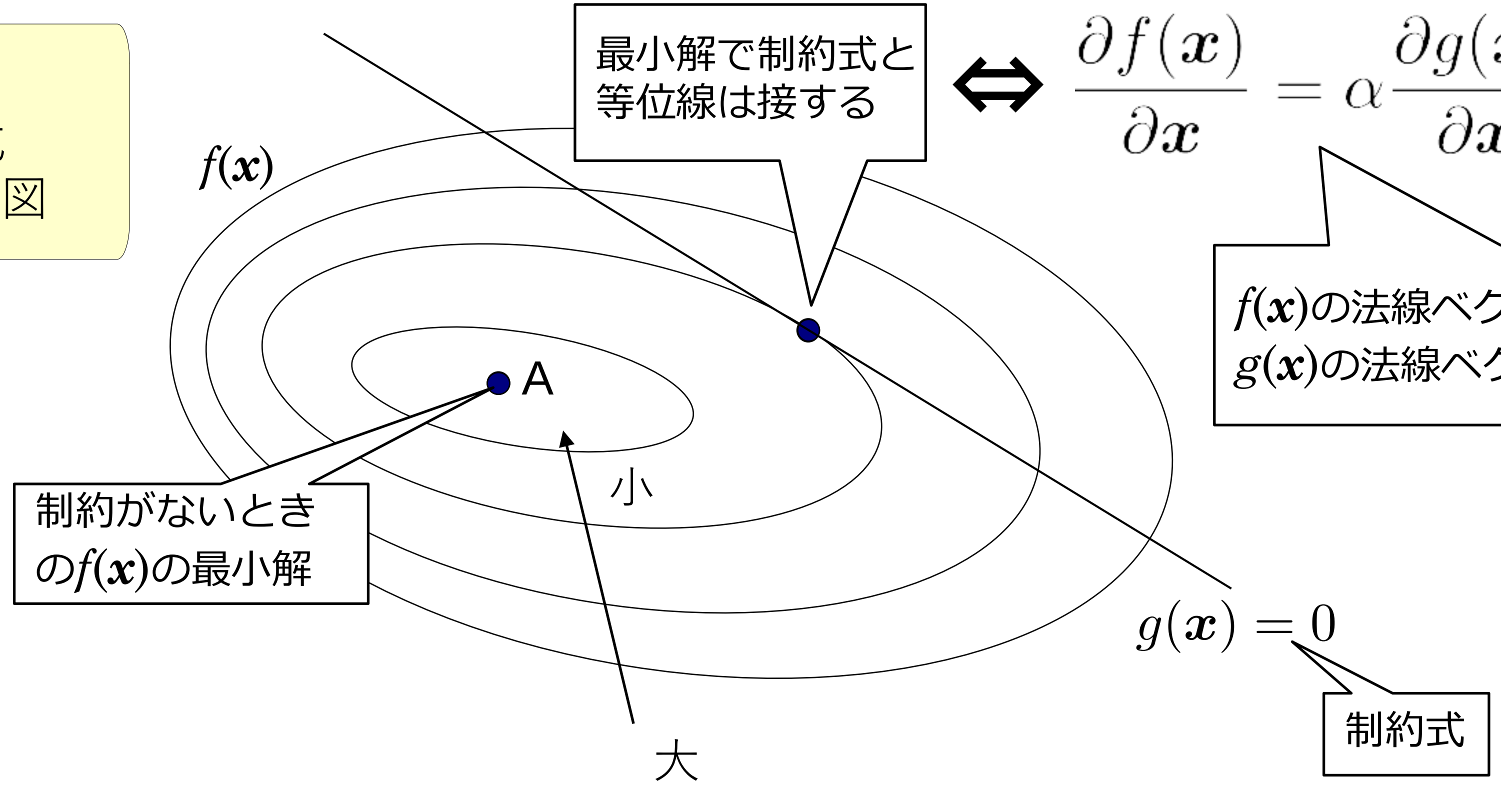
問題 $\min f(\boldsymbol{x}) \quad s.t.\ g(\boldsymbol{x}) = 0$ $\Longrightarrow$ $L(\boldsymbol{x}, \alpha) = f(\boldsymbol{x}) - \alpha g(\boldsymbol{x})$

$$\frac{\partial L(\boldsymbol{x}, \alpha)}{\partial \boldsymbol{x}} = \nabla f(\boldsymbol{x}) - \alpha \nabla g(\boldsymbol{x}) = 0$$

$$\Leftrightarrow \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} = \alpha \frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}$$

$f(\boldsymbol{x})$の法線ベクトルと
$g(\boldsymbol{x})$の法線ベクトルは方向が同じ

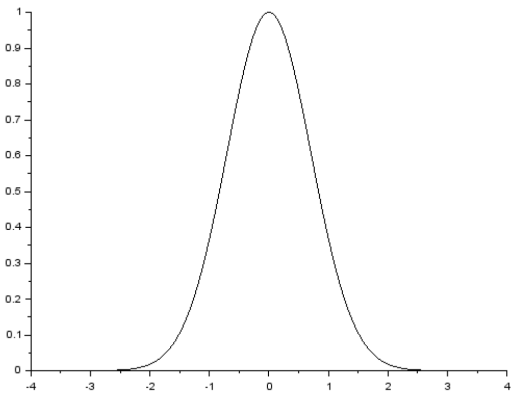
fが2次式
gが1次式
の場合の図



RBFカーネル

- RBFカーネルの解釈

$$e^{-||\boldsymbol{x}-\boldsymbol{x}'||^2}$$



$y = \exp(-x^2)$

- RBFカーネルの展開
  - ◆ $\exp(-x^2)$ のマクローリン展開

$$e^{-x^2} = 1 - x^2 + \frac{x^4}{2!} - \frac{x^6}{3!} + \frac{x^8}{4!} + \dots$$

  - ◆ RBFカーネルは無限級数の積で表されるので、無限次元ベクトルの内積と解釈できる

7.4 文書分類問題へのSVMの適用

- 文章のベクトル化
  - ◆ 例）「顔認証はヤバいぐらい便利」
    - 形態素解析：「顔認証　は　ヤバい　ぐらい　便利」

(0, …., 0, 1, 0, …., 1, 0, 1, …)

顔認証　ヤバい　便利

単語の種類数＝次元数

Positive

分類ラベル

- ◆ 高次元特徴に強いSVMを用いて識別器を学習
- ◆ 多項式カーネルを用いると単語間の共起が相関として取れるので性能が上がることもある
- ◆ ただし、元が高次元なのでむやみに次数を上げるのも危険