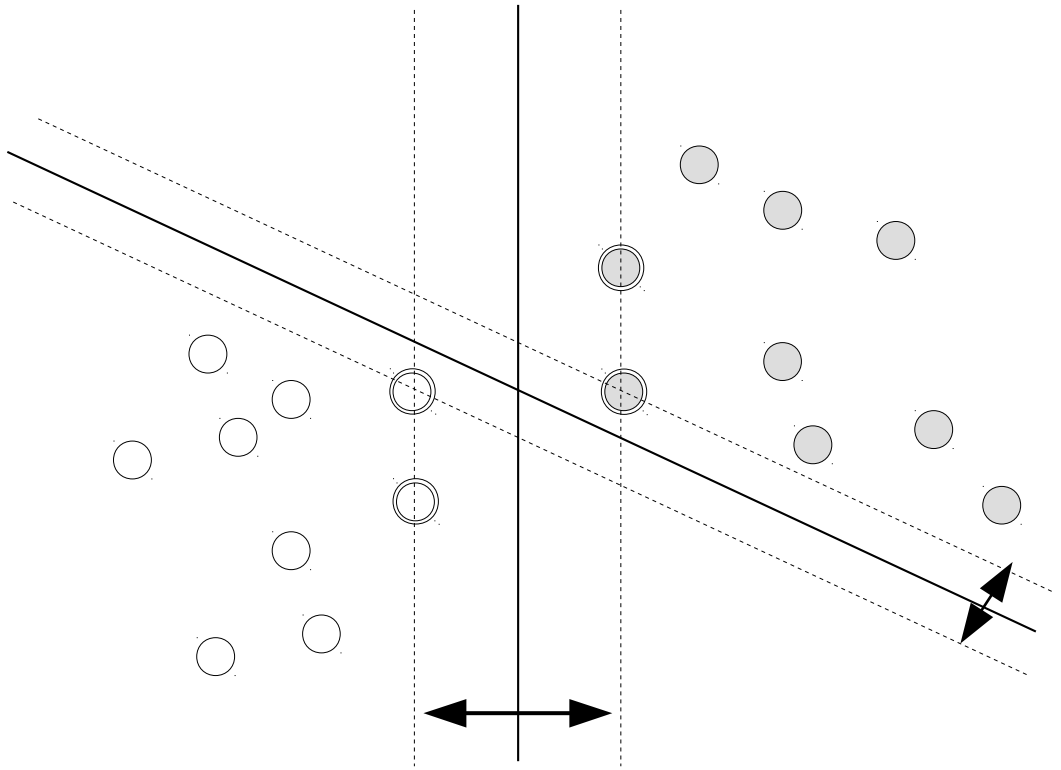
7. サポートベクトルマシン

- マージンを最大化する識別面を求める

識別面と、最も近いデータとの距離



：サポートベクトル

7.1 サポートベクトルマシンとは

- 学習データ

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

- 識別面の式

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

- 識別面の制約（係数を定数倍しても平面は不変）

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

- 学習パターンと超平面との最小距離

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

点と直線の距離の公式

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

7.1 サポートベクトルマシンとは

- 目的関数： $\min \frac{1}{2}||\boldsymbol{w}||^2$
- 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
  - 問題 $\min f(x) \quad s.t.\ g(x) = 0$
  - ラグランジュ関数 $L(x, \alpha) = f(x) + \alpha g(x)$
    - $\alpha \geqq 0$
    - x, α で偏微分して 0 になる値が極値

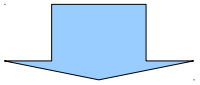
7.1 サポートベクトルマシンとは

- 計算

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

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

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



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

α についての
2 次計画問題

7.1 サポートベクトルマシンとは

- 定数項の計算
  - 各クラスのサポートベクトルから求める

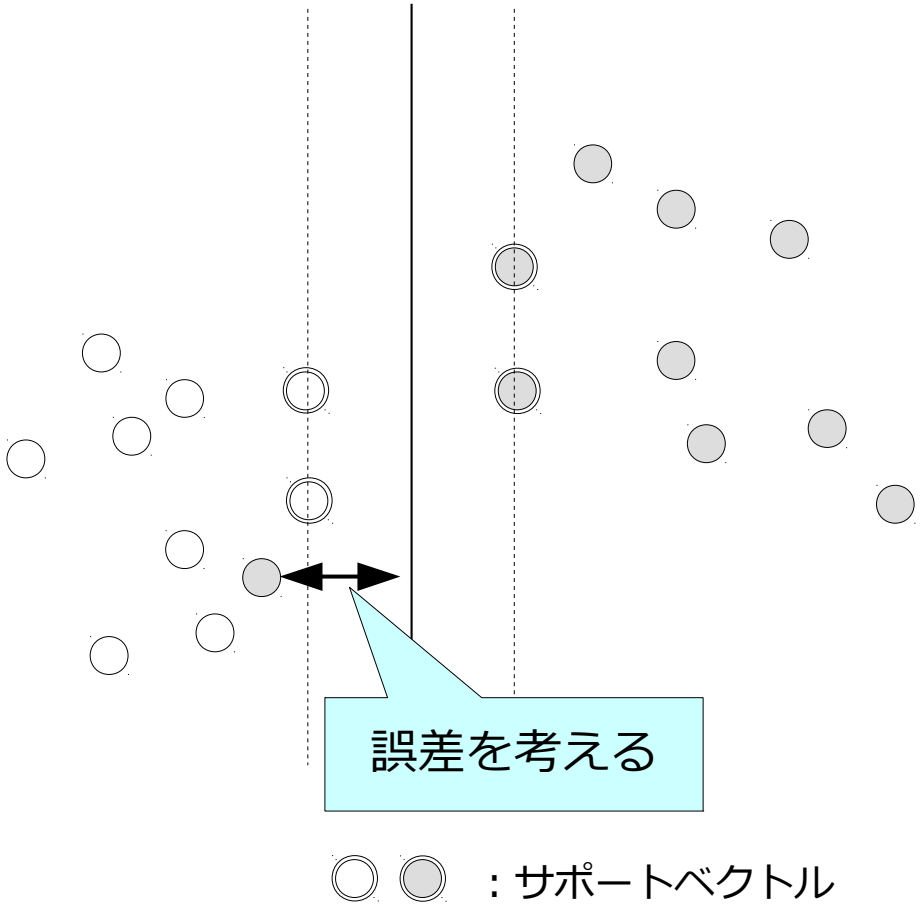
$$w_0 = -\frac{1}{2}(\boldsymbol{w}^T \boldsymbol{x}_{s1} + \boldsymbol{w}^T \boldsymbol{x}_{s2})$$

- 識別関数

$$\begin{aligned} g(\boldsymbol{x}) &= \boldsymbol{w}^T \boldsymbol{x} + w_0 \\ &= \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}^T \boldsymbol{x}_i + w_0 \end{aligned}$$

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

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

- 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

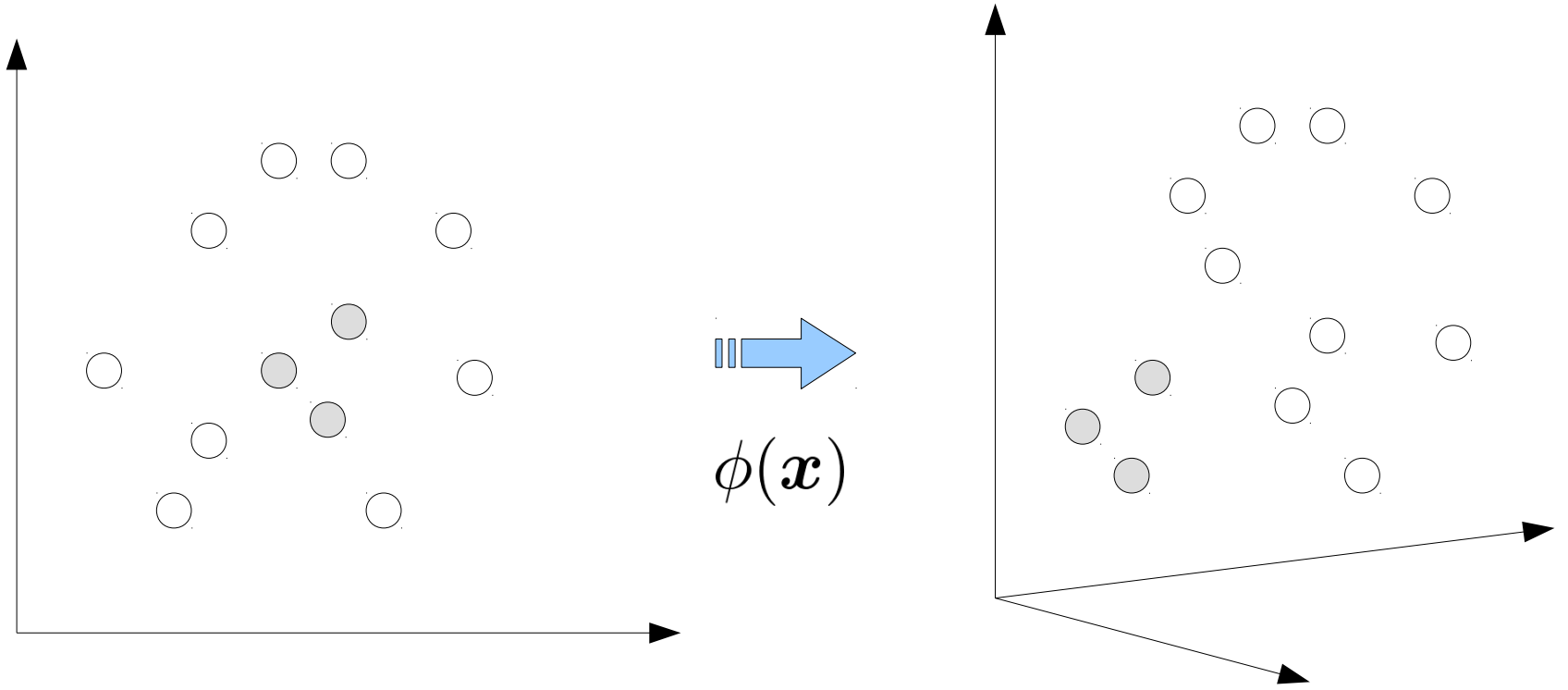
- C: エラー事例に対するペナルティ
  - 大きな値：誤識別データの影響が大きい

    → 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

- 特徴ベクトルの次元を増やす



2 次元で線形分離不可能なデータ

線形分離可能な 3 次元へ

ただし、元の空間でのデータ間の
距離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

  - 元の空間での距離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

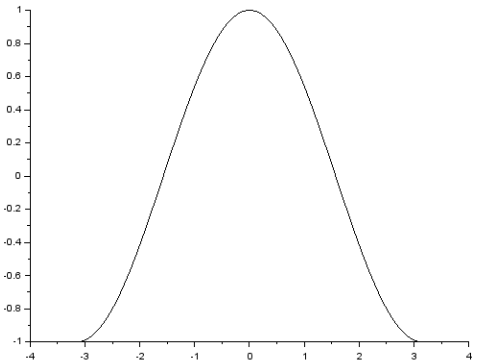
7.3 カーネル関数を用いた SVM

- カーネル関数の例（ scikit-learn の定義）
  - 線形　　$K(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^T \boldsymbol{x}'$
    - 元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
  - RBF　　$K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma ||\boldsymbol{x} - \boldsymbol{x}'||^2)$
    - γ の値：大→複雑　小→単純な識別面
  - シグモイド　$K(\boldsymbol{x}, \boldsymbol{x}') = \tanh(\boldsymbol{x}^T \boldsymbol{x}' + r)$
    - ニューラルネットワークと似た振る舞いを実現

7.3 カーネル関数を用いた SVM

- 多項式カーネルの解釈

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



$y = \cos(x)$

- 多項式カーネルの展開例

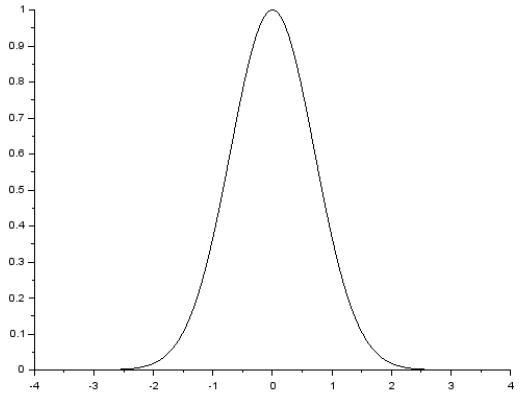
$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = (\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)} + 1)^2$$
$$= (x_1^{(i)}x_1^{(j)} + x_2^{(i)}x_2^{(j)} + 1)^2$$
$$= (x_1^{(i)}x_1^{(j)})^2 + (x_2^{(i)}x_2^{(j)})^2 + 2x_1^{(i)}x_1^{(j)}x_2^{(i)}x_2^{(j)} + 2x_1^{(i)}x_1^{(j)} + 2x_2^{(i)}x_2^{(j)} + 1$$
$$= ((x_1^{(i)})^2, (x_2^{(i)})^2, \sqrt{2}x_1^{(i)}x_2^{(i)}, \sqrt{2}x_1^{(i)}, \sqrt{2}x_2^{(i)}, 1)$$
$$\cdot ((x_1^{(j)})^2, (x_2^{(j)})^2, \sqrt{2}x_1^{(j)}x_2^{(j)}, \sqrt{2}x_1^{(j)}, \sqrt{2}x_2^{(j)}, 1)$$

2 変数の相関を表す項

7.3 カーネル関数を用いた SVM

- RBF カーネルの解釈

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



- RBF カーネルの展開
  - e^{-x2} のマクローリン展開 (maclaurin expansion)

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

  - ガウシアン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

7.3 カーネル関数を用いた SVM

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

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

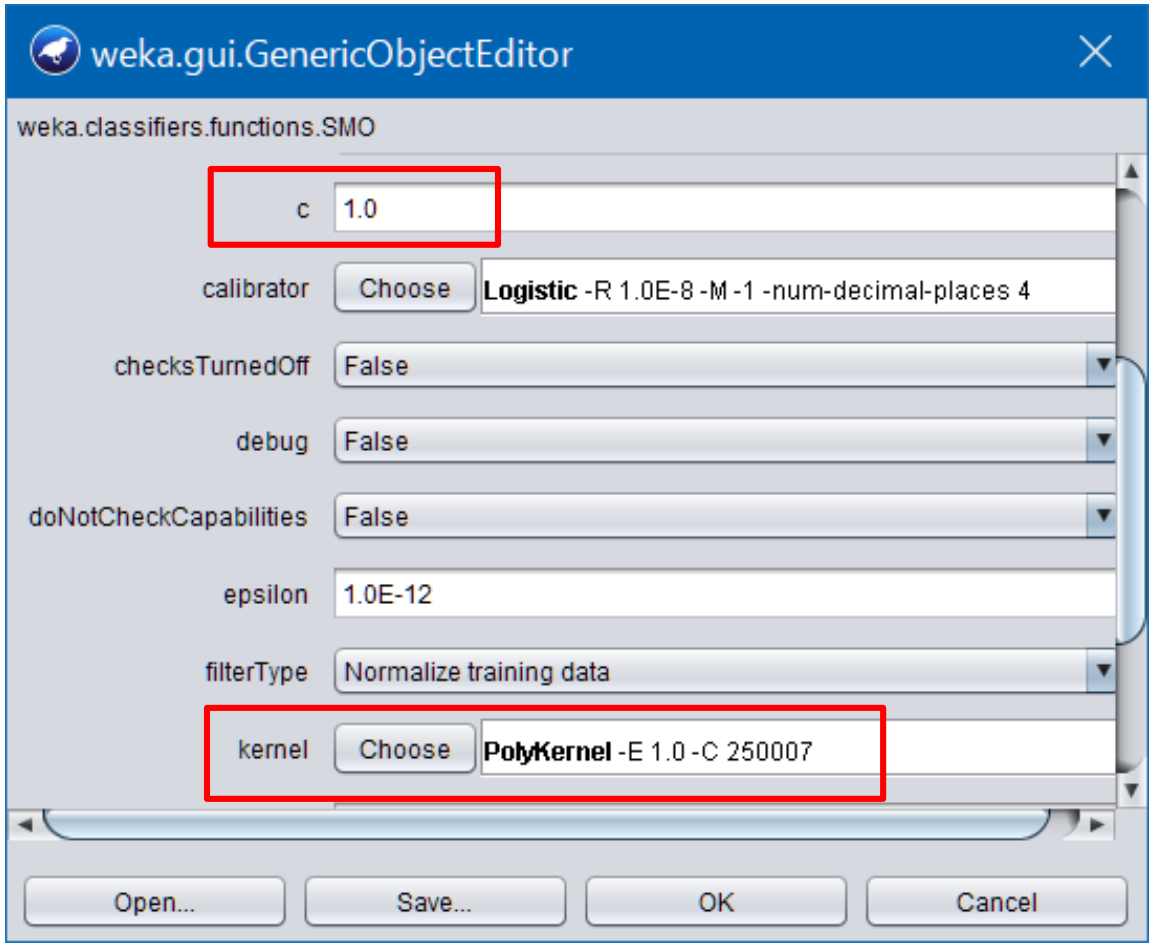
$$= \sum_{i=1}^{N} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + w_0$$

非線形変換の
式は不要！！！

カーネルトリック

7.3 カーネル関数を用いた SVM

- Weka でのパラメータ調整



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

7.4 文書分類問題へのSVMの適用

- SVMによる評判分析
  - 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

Positive

分類ラベル

単語の種類数
＝次元数

  - 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

サポートベクトル回帰

- 基底関数にカーネルを用いる

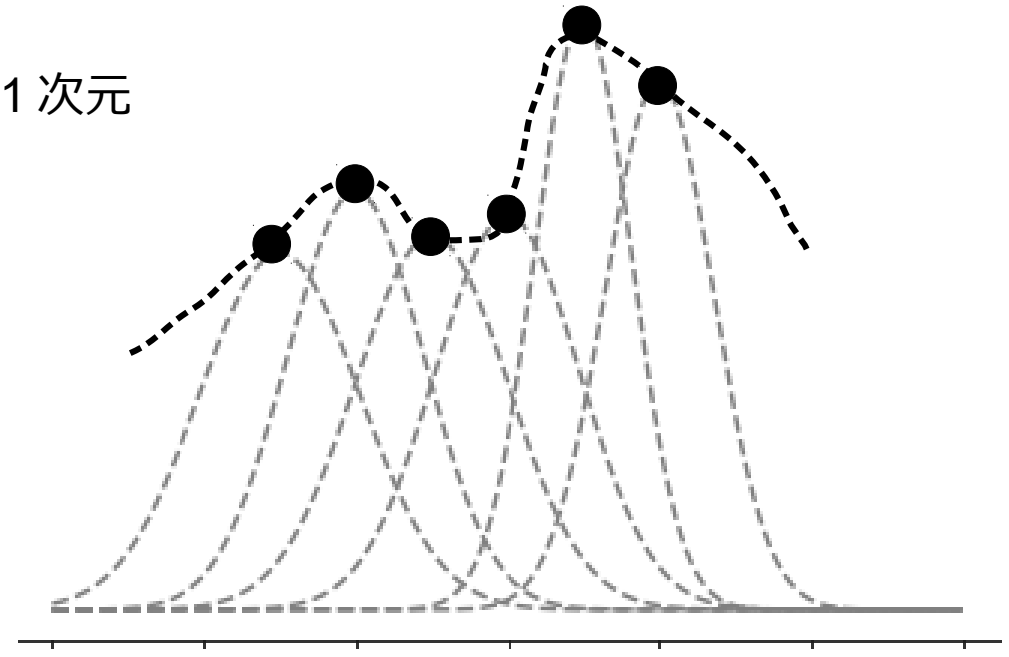
$$\hat{c}(\boldsymbol{x}) = \sum_{j=1}^{N} \alpha_j K(\boldsymbol{x}, \boldsymbol{x}_j)$$

- RBF カーネルを用いた場合

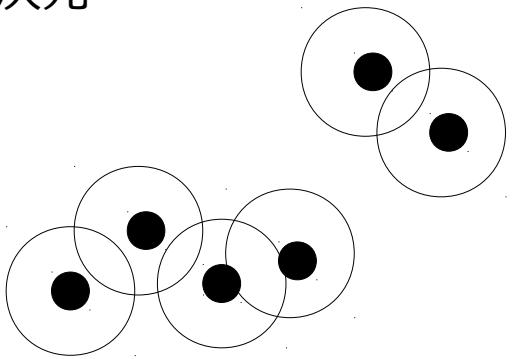
$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma ||\boldsymbol{x} - \boldsymbol{x}'||^2)$$

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



サポートベクトル回帰

- 学習データ

$$\chi = \{(\boldsymbol{x}_i, y_i)\} \quad i = 1, \ldots, N, \quad y_i \in \mathbb{R}$$

- 最適化対象

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

$$y_i - \boldsymbol{w}^T\phi(\boldsymbol{x}_i) - w_0 \leq \epsilon + \xi_i$$

$$\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + w_0 - y_i \leq \epsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0 \quad i = 1, \ldots, N$$

- ラグランジュの未定乗数法で双対問題に変換し、 α を求める
- 高次元変換 Φ が制約式に出現しているが、双対問題に変換すればカーネル式となり、消える