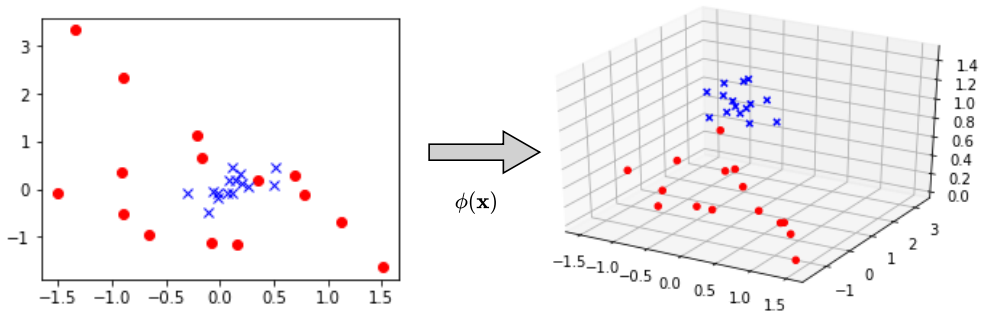
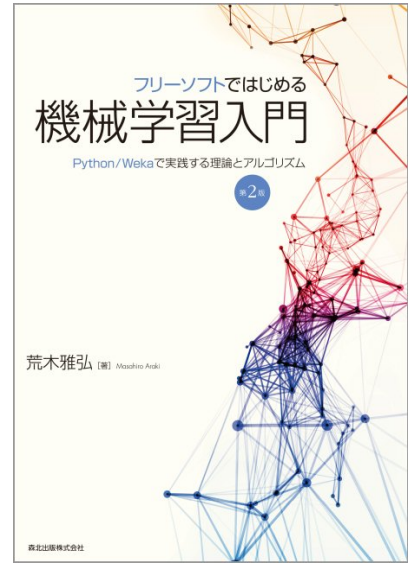
7. サポートベクトルマシン



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写像

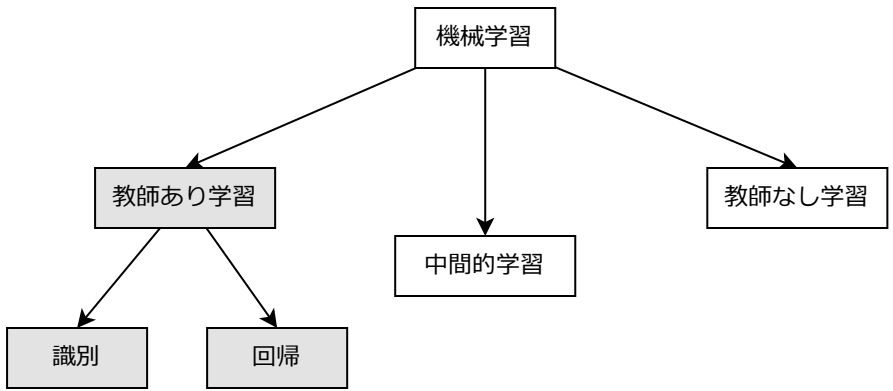
- 7.1 サポートベクトルマシンとは
- 7.2 ソフトマージンによる誤識別データの吸収
- 7.3 カーネル関数を用いたSVM
- 7.4 文書分類問題へのSVMの適用
- 7.5 ハイパーパラメータの調整



- 荒木雅弘:『フリーソフトではじめる機械学習入門(第2版)』(森北出版, 2018年)
- スライドとJupyter notebook
- サポートページ

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
- 線形モデルでは高い性能が得られないデータに適する

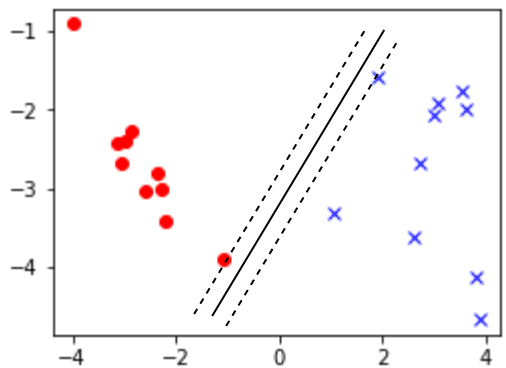


7. サポートベクトルマシン

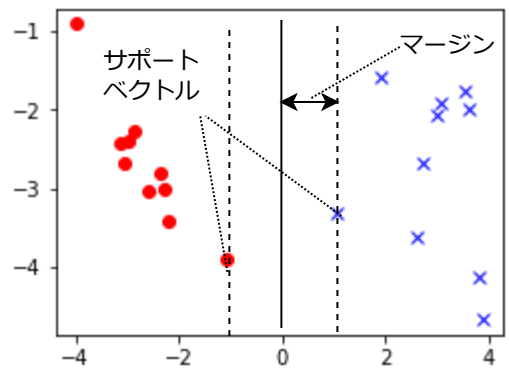
- 本章の説明手順
  1. 線形分離可能なデータに対して、なるべく学習データに特化しすぎない識別面を求める
  2. 線形分離不可能なデータに対して、誤識別に対するペナルティを設定することで、1. の手法を改良する
  3. さらに複雑なデータに対して、学習データを高次元の空間に写して、2. の手法を適用する
  4. 3.の手法に対して、最適なハイパーパラメータを求める

7.1 サポートベクトルマシンとは（1/6）

- 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：識別面と最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 サポートベクトルマシンとは（2/6）

- マージン最大化のための定式化
  - 学習データ: $\boldsymbol{x} \in \mathbb{R}^d$, $y \in \{-1, 1\}$

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

  - 識別面の式（超平面）

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

  - 識別面の式に制約を導入（係数を定数倍しても超平面は不変）

$$\min_i |\boldsymbol{w}^T \boldsymbol{x}_i + w_0| = 1$$

7.1 サポートベクトルマシンとは（3/6）

- 学習対象の設定
  - 学習データと識別面との最小距離（マージン）　cf）点と直線の距離の公式

$$\min_i \mathrm{Dist}(\boldsymbol{x}_i) = \min_i \frac{|\boldsymbol{w}^T \boldsymbol{x}_i + w_0|}{\|\boldsymbol{w}\|} = \frac{1}{\|\boldsymbol{w}\|}$$

  - 目的関数：マージン最大化を、逆数の2乗の最小化と置き換える

$$\min \frac{1}{2}\|\boldsymbol{w}\|^2$$

    - 制約条件：$\boldsymbol{w}$ で定まる超平面で正しく識別が行えること

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

7.1 サポートベクトルマシンとは（4/6）

- 最適化問題の解法：ラグランジュの未定乗数法
  - 例題（2変数、等式制約）

$$\min f(x,y) \;\; s.t.\; g(x,y) = 0$$

  - ラグランジュ関数の導入：$L(x,y,\alpha) = f(x,y) + \alpha g(x,y)$
  - ラグランジュ係数の制約：$\alpha \geq 0$
  - L を x, y, α で偏微分して 0 になる値が f の極値
    - 変数3つに対して制約式が3つなので、変数について解ける

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

7.1 サポートベクトルマシンとは（5/6）

- より解きやすい問題への変換

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

$$\frac{\partial L}{\partial w_0} = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0, \qquad \frac{\partial L}{\partial \boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i$$

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

  - 最後の式の最小化は $\alpha_i \geq 0$ での2次計画問題なので、容易に極値となる $\boldsymbol{\alpha}$ を求めることができ、そこから $\boldsymbol{w}$ が求まる

7.1 サポートベクトルマシンとは（6/6）

- 定数項の計算
  - 各クラスのサポートベクトルを引数とした識別関数の値の絶対値が1であることから求める

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

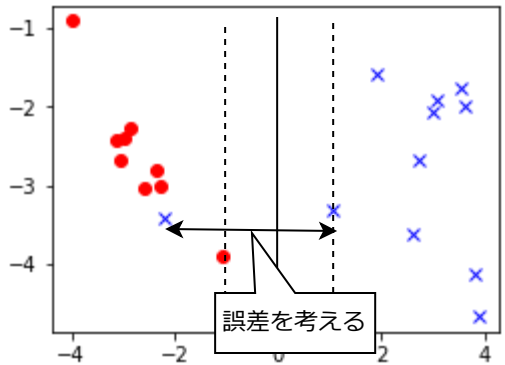
- マージンが最大の識別関数

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

  - サポートベクトル $\boldsymbol{x}_i$ に対応する α_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（1/3）

- 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（2/3）

- スラック変数 ξ_i の導入

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

  - $\xi_i = 0$：マージンの外側、$0 < \xi_i \leq 1$：マージンと識別面の間、$1 < \xi_i$：誤り
- 最小化問題の修正: スラック変数も小さい方がよい

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

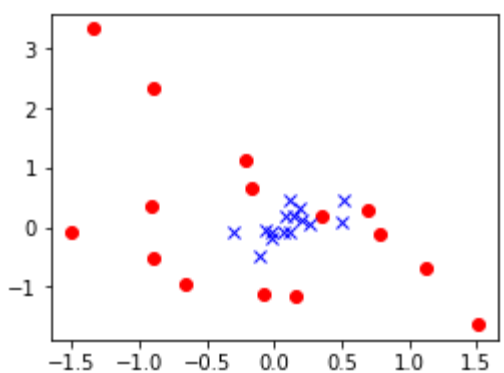
- 計算結果
  - α_i の2次計画問題に $0 \leq \alpha_i \leq C$ が加わるだけ

7.2 ソフトマージンによる誤識別データの吸収（3/3）

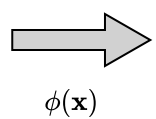
- C：エラー事例に対するペナルティの大きさ
  - 大きな値：誤識別データの影響が大きい
    - マージンが狭くても誤識別をなるべく減らすようにする
  - 小さな値：誤識別データの影響が小さい
    - 多少の誤識別があっても、なるべくマージンを広くする

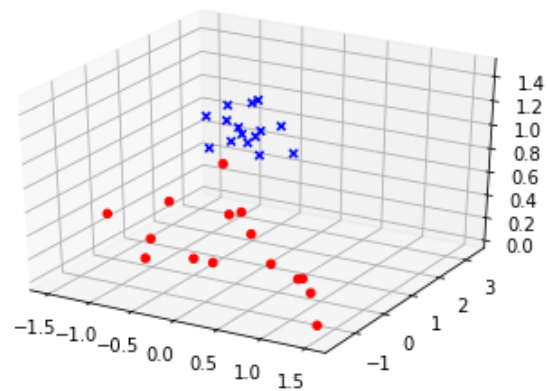
7.3 カーネル関数を用いたSVM（1/7）

- 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して線形分離可能性を高める
  - 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写像する



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写像

7.3 カーネル関数を用いたSVM（2/7）

- 高次元への非線形変換関数 $\boldsymbol{\phi}(\boldsymbol{x})$ を設定する
- カーネル関数：2つの点の近さを表す

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

  - もとの空間での2つの点の近さを、変換後の空間での2つのベクトルの内積に対応させる
  - $\boldsymbol{x}$ と $\boldsymbol{x}'$ が近ければ $k(\boldsymbol{x}, \boldsymbol{x}')$ は大きい値
  - カーネル関数が正定値性を満たすとき、このような非線形変換が存在することが保証されている

7.3 カーネル関数を用いたSVM（3/7）

- 正定値カーネル関数の例（scikit-learn SVCの定義）
  - 線形（linear）：$k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^T \boldsymbol{x}'$
    - 高次元に写像せず、もとの特徴空間でマージン最大の超平面を求めるときのカーネル関数
  - 多項式（poly）：$k(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^T \boldsymbol{x}' + r)^d$
    - d 項の相関を加えるので d が大きいほど複雑な識別面。r はたいてい1
  - ガウシアン（rbf）：$k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{x}'\|^2)$
    - γ が大きいほど近くのデータしか影響しないので複雑な識別面
  - シグモイド（sigmoid）：$k(\boldsymbol{x}, \boldsymbol{x}') = \tanh(\boldsymbol{x}^T \boldsymbol{x}' + r)$
    - 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 (4/7)

- 線形カーネルの解釈
  - θ が 0 に近い(＝ベクトルとして似ている)ほど大きな値

$$K(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^T \boldsymbol{x}' = \|\boldsymbol{x}\| \cdot \|\boldsymbol{x}'\| \cdot \cos\theta$$

- 2次多項式カーネルの展開例($\boldsymbol{x}$ が2次元の場合 $\rightarrow$ 6次元空間に写像されている)

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

7.3 カーネル関数を用いたSVM（5/7）

- RBFカーネルの解釈

$$
\begin{aligned}
K(\boldsymbol{x}, \boldsymbol{x}') &= \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{x}'\|^2) = \exp(-\gamma(\|\boldsymbol{x}\|^2 - 2\boldsymbol{x}^T\boldsymbol{x}' + \|\boldsymbol{x}'\|^2)) \\
&= \exp(-\gamma\|\boldsymbol{x}\|^2)\exp(-\gamma\|\boldsymbol{x}'\|^2)\exp(2\gamma\boldsymbol{x}^T\boldsymbol{x}')
\end{aligned}
$$

- RBFカーネルの展開
  - $\exp(x)$ のマクローリン展開

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

  - RBFカーネルを展開した第3項は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（6/7）

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

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

  - カーネルトリック：非線形変換の式は不要で、カーネル関数を用いて元の空間の情報のみで識別関数が書ける
- 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（7/7）

- sklearn で識別を行う SVM の定義

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

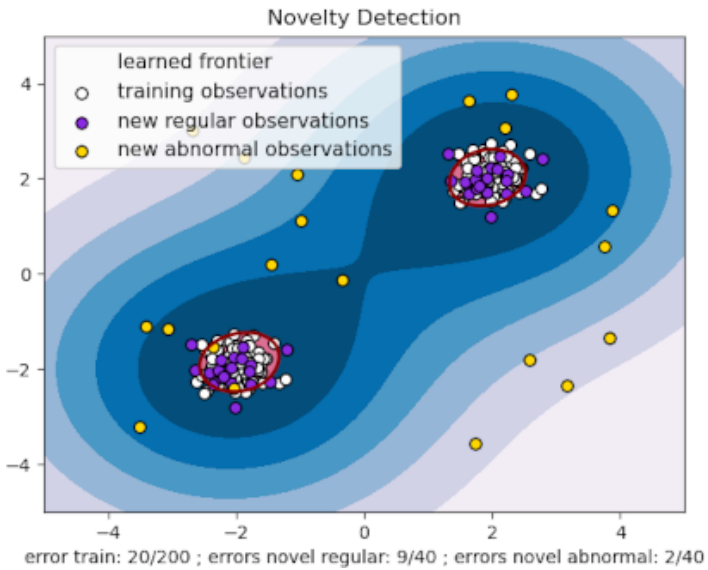
- ハイパーパラメータ
  - C：スラック変数の重み
  - kernel：カーネルの種類
    - ’poly’：多項式カーネル
    - ’rbf’：RBFカーネル
    - その他：’linear’, ’sigmoid’, ’precomputed’
  - degree：polyカーネルを指定したときの次数
  - gamma：（主として）RBFカーネルを指定したときの係数

7.4 文書分類問題へのSVMの適用

- 文章のベクトル化
  - 入力例)「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」
    - 単語ベクトル化：$(0, 0, \ldots, 0, 1, 0, \ldots, 0, 1, 0, \ldots)$
      - 各次元が単語辞書の見出し順に対応し、入力に出現した単語の次元の値が1、それ以外は0
- 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

One-class SVM

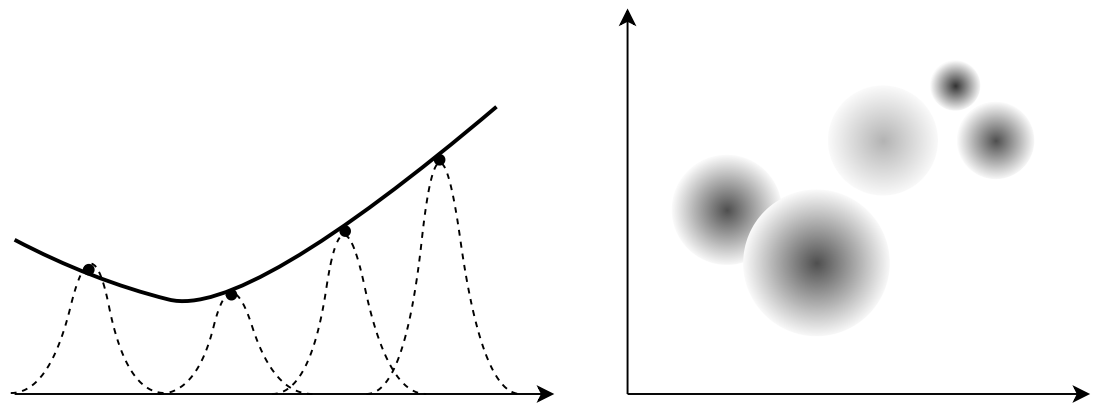
- One-class SVMによる新規検知
  - RBFカーネルによる写像後の学習データを正例、原点を負例とみなして境界を得る
  - 新規データに対して、境界の外の場合は異常とみなす



サポートベクトル回帰

- 回帰における基底関数としてカーネル関数を用いる
  - RBFカーネルを用いた場合

$$\hat{c}(\boldsymbol{x}) = \sum_{j=1}^{N} \alpha_j K(\boldsymbol{x}, \boldsymbol{x}_j) = \sum_{j=1}^{N} \alpha_j \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{x}_j\|^2)$$



1次元特徴の場合　　2次元特徴の場合

7.5 ハイパーパラメータの調整 (1/2)

- パラメータ：学習データを用いて調整
  - 線形識別関数の重み $\boldsymbol{w}$
  - SVMの $\boldsymbol{\alpha}$
  - 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もの
  - 高次識別関数の次数 b、正則化項の重み α
  - SVM におけるスラック変数の重み 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調整（2/2）

- 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
    - 格子点の最小値・最大値・間隔等の知見が必要
    - ハイパーパラメータが連続値の場合は等比数列で設定する
  - ランダムサーチ：各ハイパーパラメータの適切な範囲内で乱数によって複数点を設定
    - 性能に大きく寄与するハイパーパラメータが存在する場合に有効
  - ベイズ最適化
    - ハイパーパラメータの値を入力、性能を出力とした回帰問題を設定し、ガウス過程回帰を用いて性能が高くなりそうなハイパーパラメータを探索

7.6 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、カーネルトリックによって元の空間での情報のみで識別関数を表現できる
- 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ハイパーパラメータチューニングを行う

ベイズ最適化（1/4）

- ガウス過程とは
  - ランダムな関数を生成する確率分布
  - 平均が異なる正規分布の重み付き和で回帰関数を表現

$$y = \sum_{h=0}^{H} w_h \exp\left(-\frac{(x-\mu_h)^2}{\sigma^2}\right)$$

    - x がハイパーパラメータ、y が性能
  - この関数に対応する基底関数

$$\boldsymbol{\phi}(x) = (\phi_0(x), \phi_1(x), \ldots, \phi_H(x))^T$$

ベイズ最適化（2/4）

- 線形回帰モデル

$$\boldsymbol{y} = \boldsymbol{\Phi w}$$

  - $\boldsymbol{\Phi}$：基底関数適用後のパターン行列
  - 重み $\boldsymbol{w}$ が平均 $\mathbf{0}$、分散 $\lambda^2 \boldsymbol{I}$ の正規分布から生成されたと仮定すると、その線形変換である $\boldsymbol{y}$ も正規分布
    - 平均：$E[\boldsymbol{y}] = E[\boldsymbol{\Phi w}] = \boldsymbol{\Phi} E[\boldsymbol{w}] = \mathbf{0}$
    - 分散：$\Sigma = E[\boldsymbol{y}\boldsymbol{y}^T] - E[\boldsymbol{y}]E[\boldsymbol{y}]^T = E[(\boldsymbol{\Phi w})(\boldsymbol{\Phi w})^T] = \boldsymbol{\Phi} E[\boldsymbol{w}] E[\boldsymbol{w}^T] \boldsymbol{\Phi}^T = \lambda^2 \boldsymbol{\Phi\Phi}^T$
  - $\boldsymbol{y}$ の分布

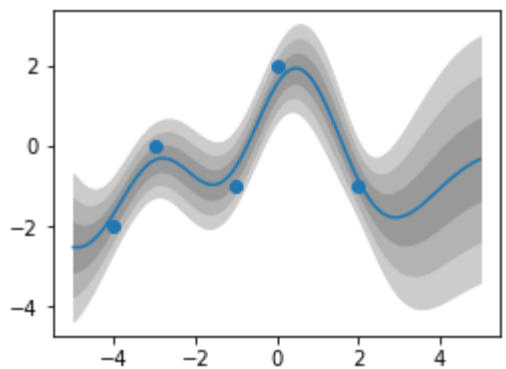
$$\boldsymbol{y} \sim \mathcal{N}(\mathbf{0}, \lambda^2 \boldsymbol{\Phi\Phi}^T)$$

ベイズ最適化（3/4）

- カーネルを用いたガウス過程の表現

$$\boldsymbol{K} = \lambda^2 \boldsymbol{\Phi}\boldsymbol{\Phi}^T$$

  - $\boldsymbol{K}$ はN行N列で、n行n’列の要素は $K_{nn'} = \boldsymbol{\phi}(\boldsymbol{x}_n)^T \boldsymbol{\phi}(\boldsymbol{x}_{n'})$
  - 回帰関数がカーネル関数を用いた多次元正規分布で表現できた



ベイズ最適化（4/4）

- ベイズ最適化の手順
  - ハイパーパラメータを特徴ベクトル、評価値をターゲットと見なしてガウス過程回帰を行う
  - 最適値になりそうな近辺を中心にハイパーパラメータを探索