機械学習
サポートベクターマシン

管理工学科
篠沢佳久

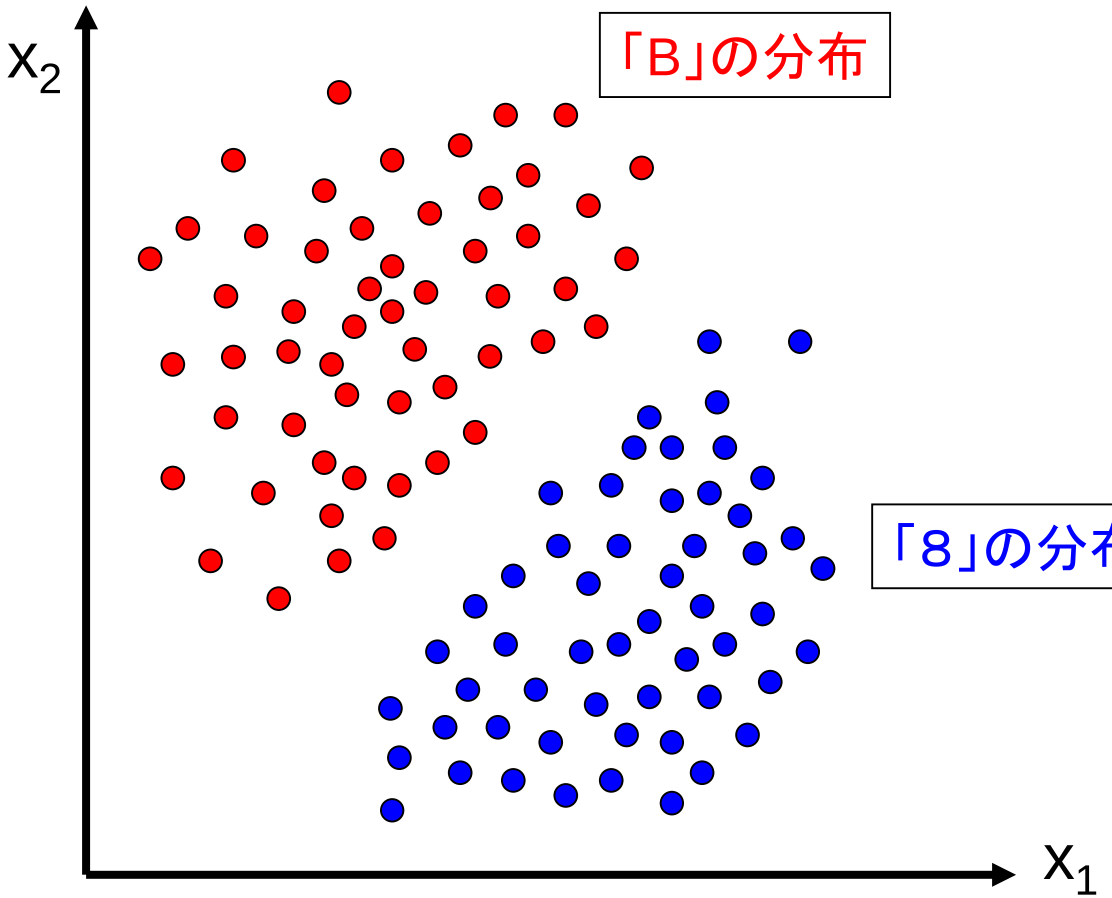
資料の内容

- サポートベクターマシン（SVM：Support Vector Machine）
  - サポートベクター
  - ハードマージン，ソフトマージン
  - カーネル法
- 実習
  - クラス分類：Support Vector Classification（SVC）
  - 回帰：Support Vector Regression（SVR）

サポートベクターマシン

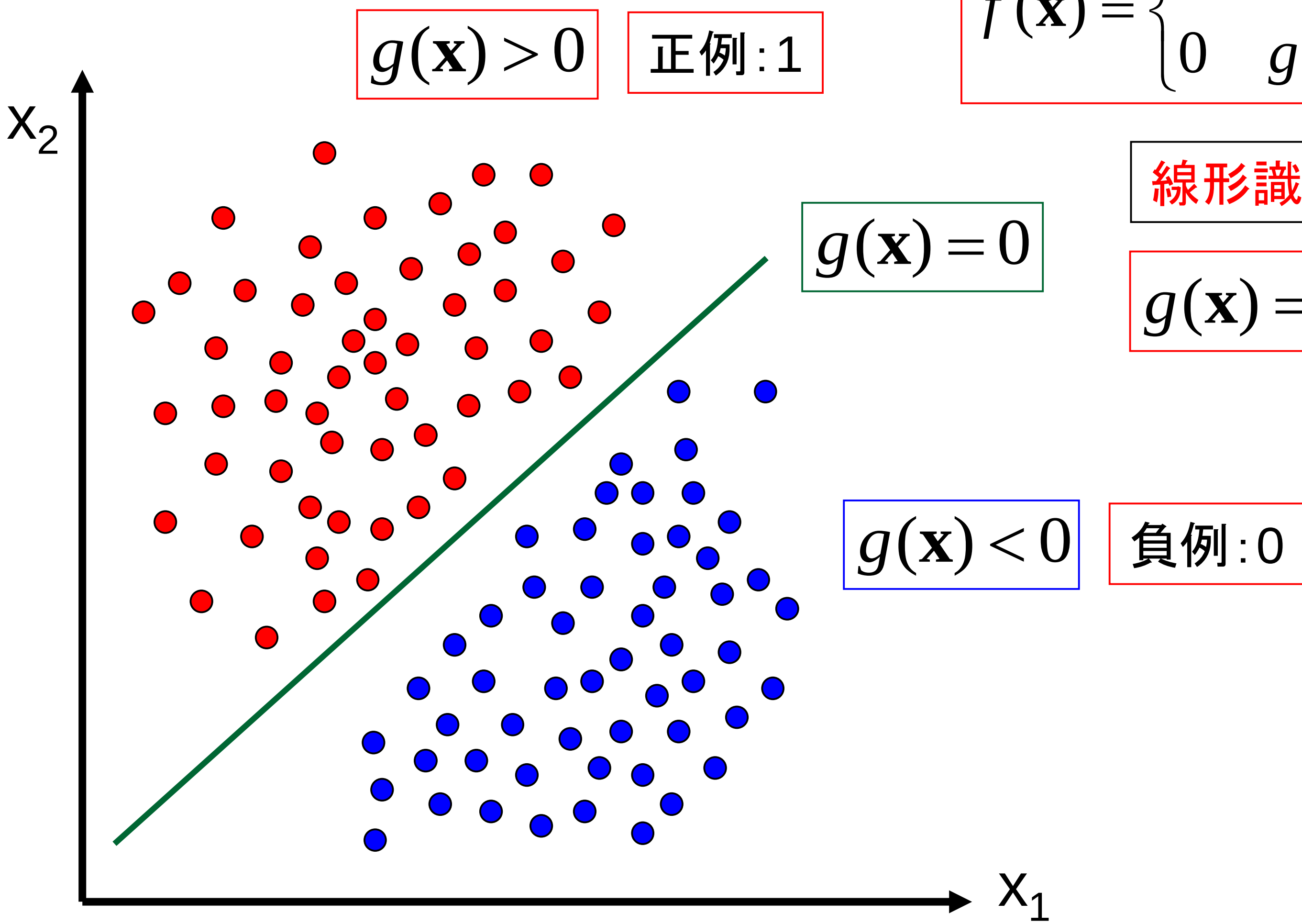
特徴空間(復習)

- 「8」と「B」の特徴ベクトル(x_1, x_2)を平面上に表現



線形識別関数による二値分類

$$f(\mathbf{x}) = \begin{cases} 1 & g(\mathbf{x}) > 0 \\ 0 & g(\mathbf{x}) < 0 \end{cases}$$

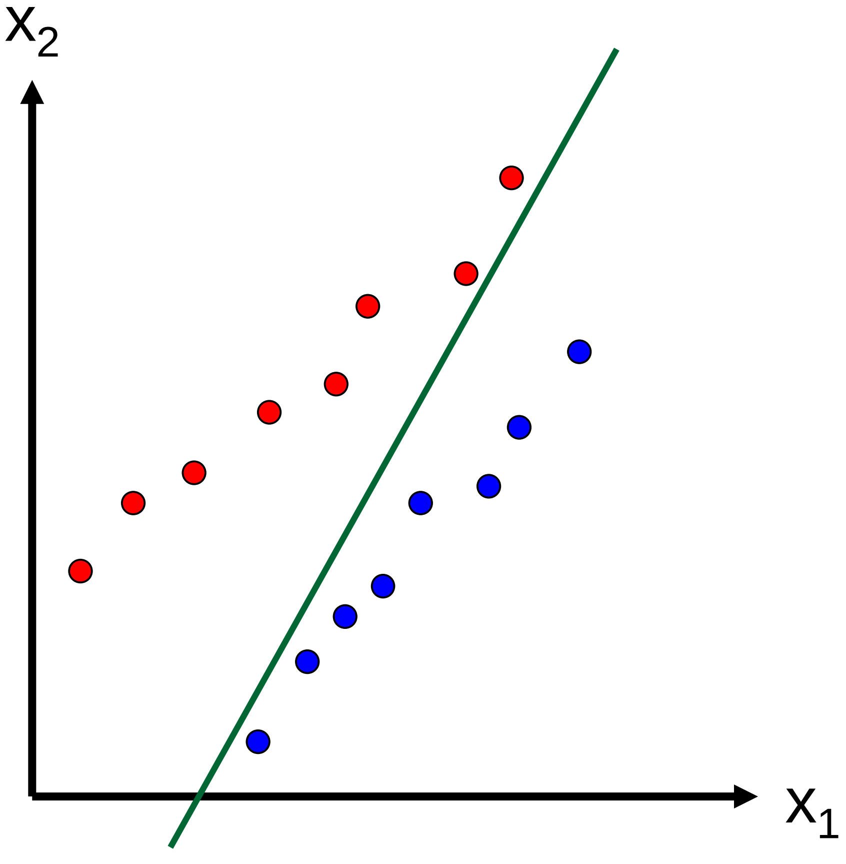


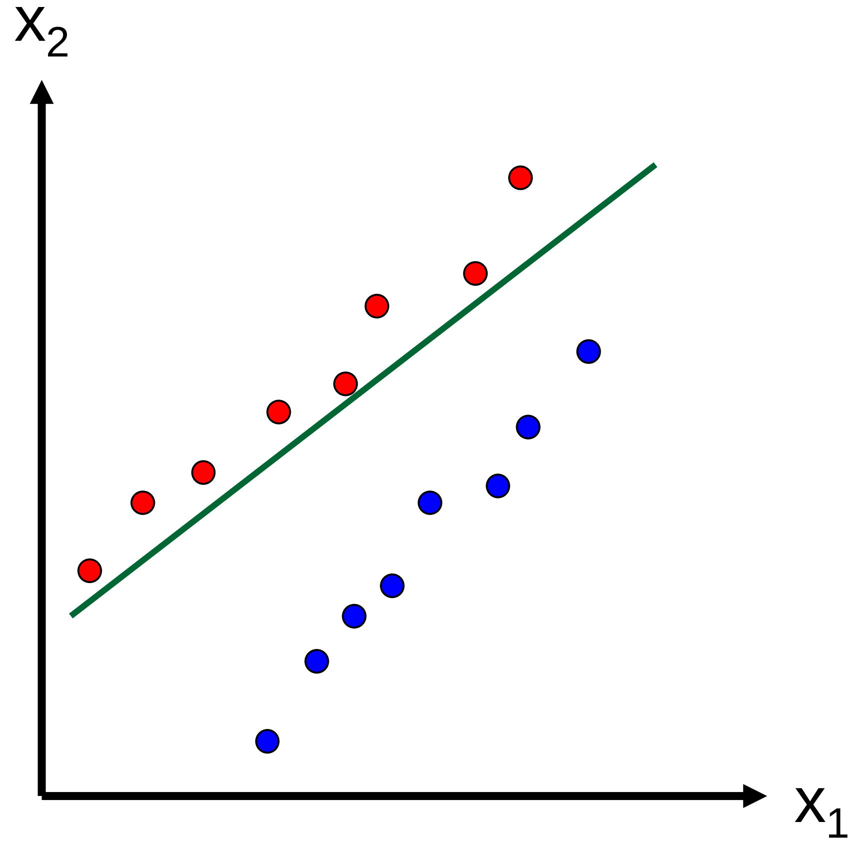
線形識別関数

$$g(\mathbf{x}) = \mathbf{w}^t \mathbf{x} + b$$

境界（$g(\mathbf{x})=0$）の引き方①

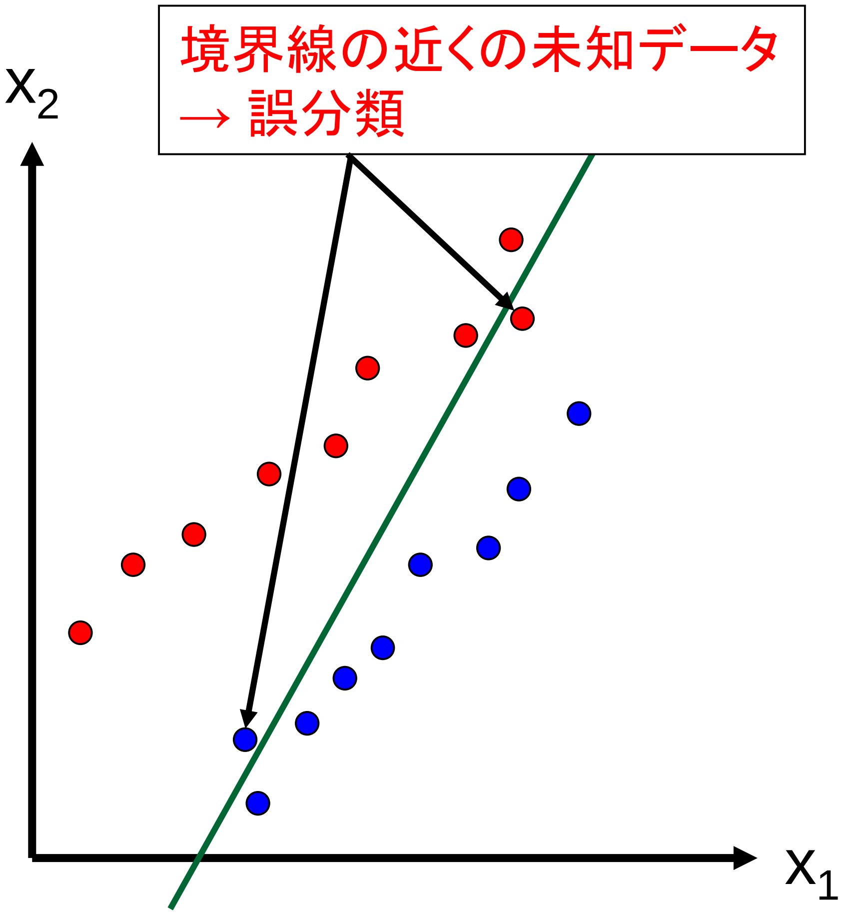
線形分離可能な場合，パーセプトロンにおいては，
無数の境界を引くことが可能

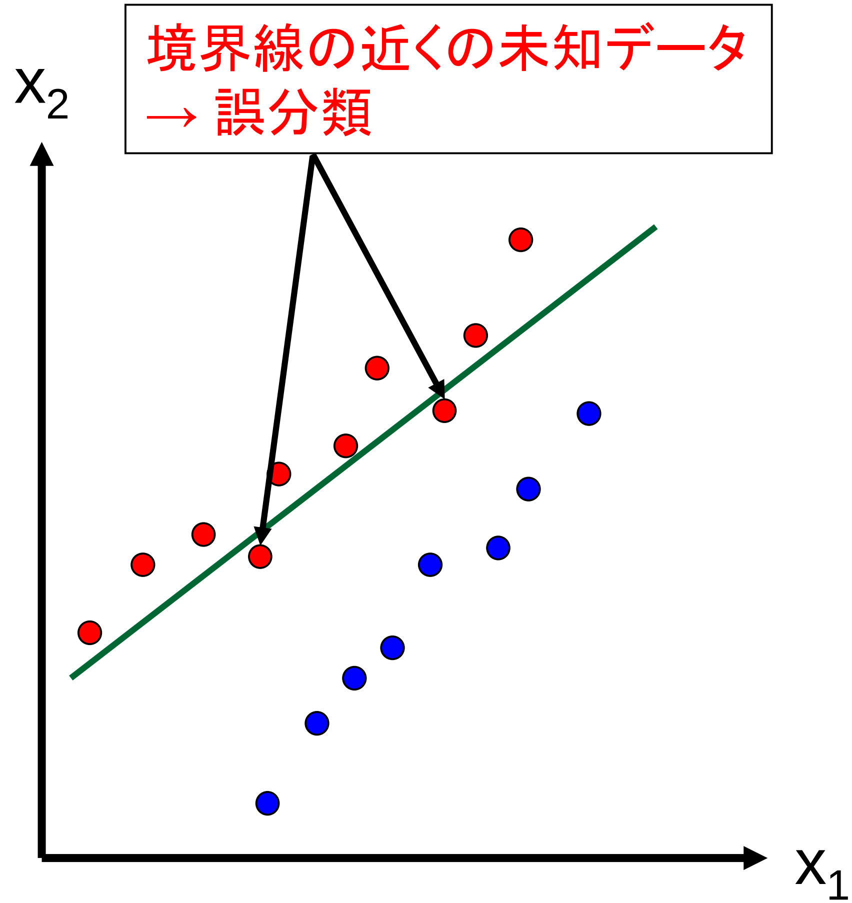




境界（g($\mathbf{x}$)=0）の引き方②

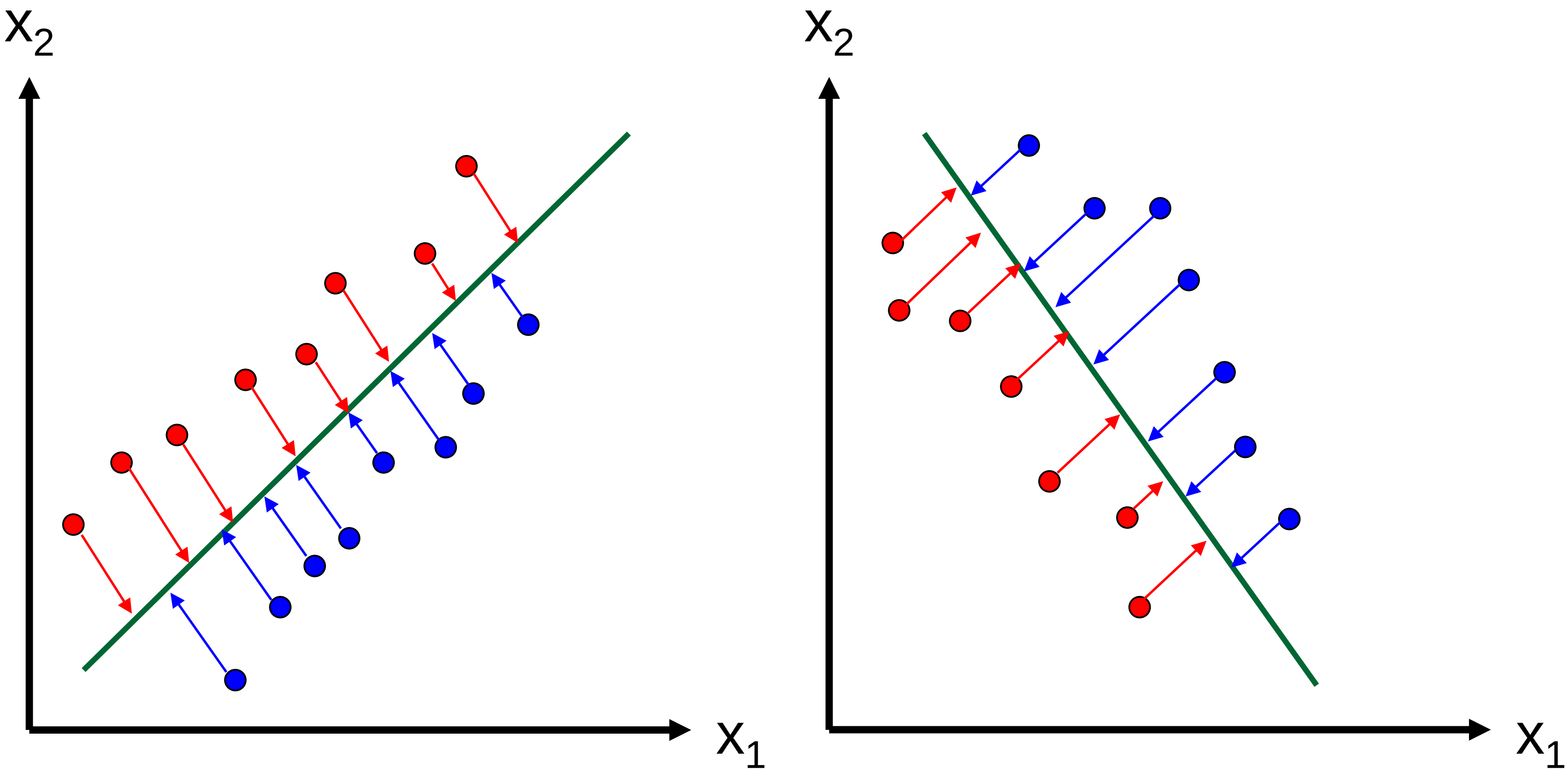
問題点



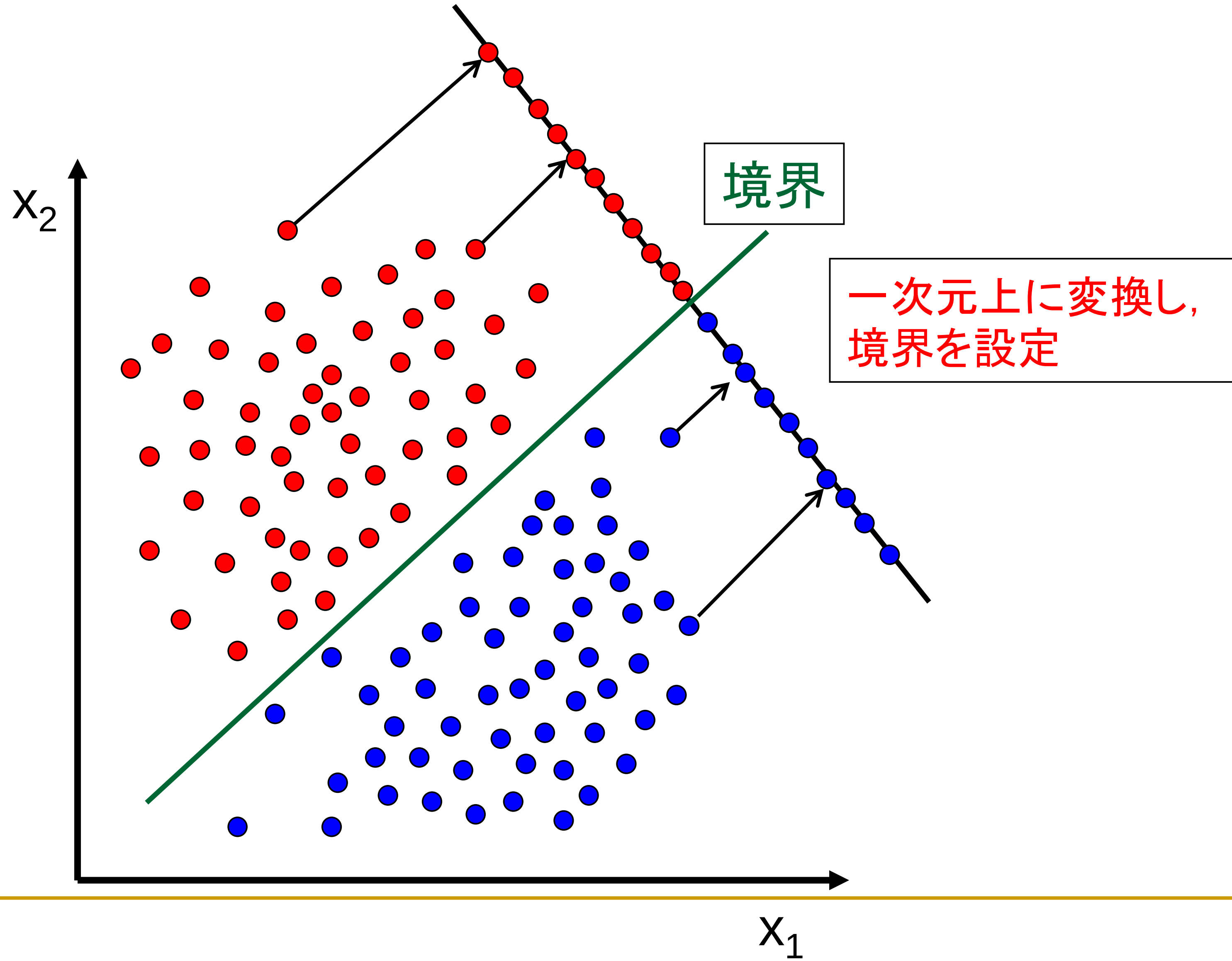


境界($g(\mathbf{x})=0$)の引き方③

境界を可能な限り全ての点から離す

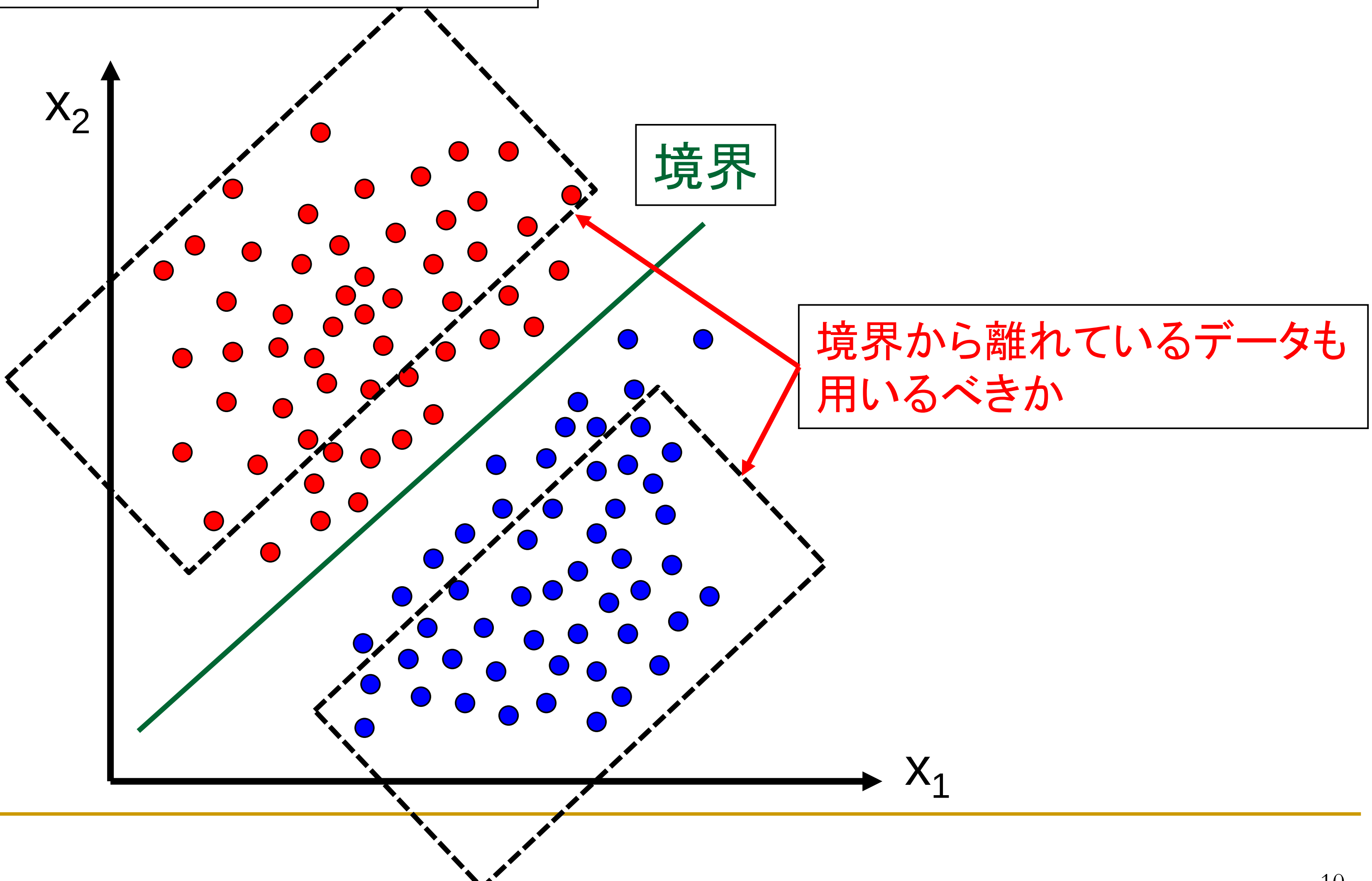


境界($g(x)=0$)の引き方④



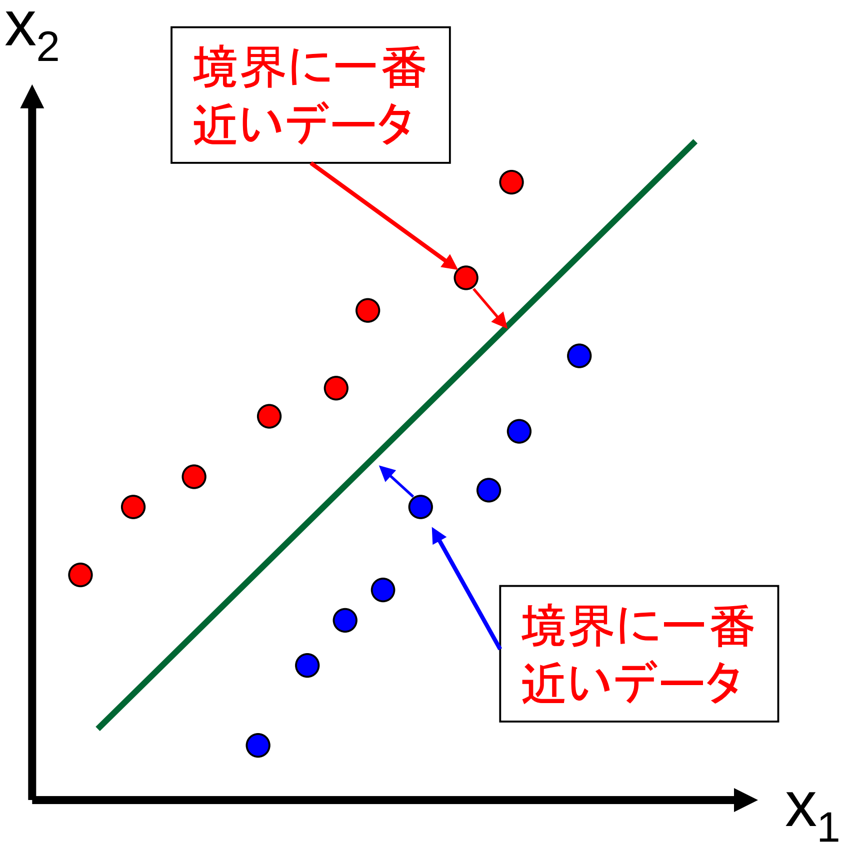
境界(g(x)=0)の引き方⑤

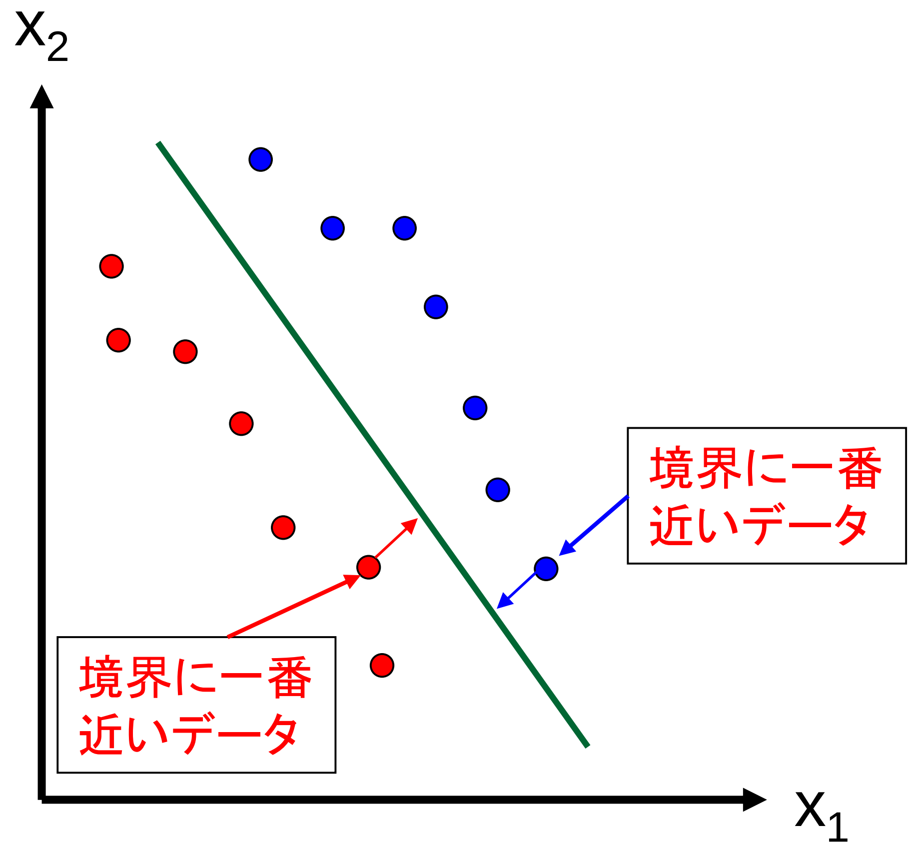
全データを考慮すべきか



サポートベクターマシンにおける境界の引き方①

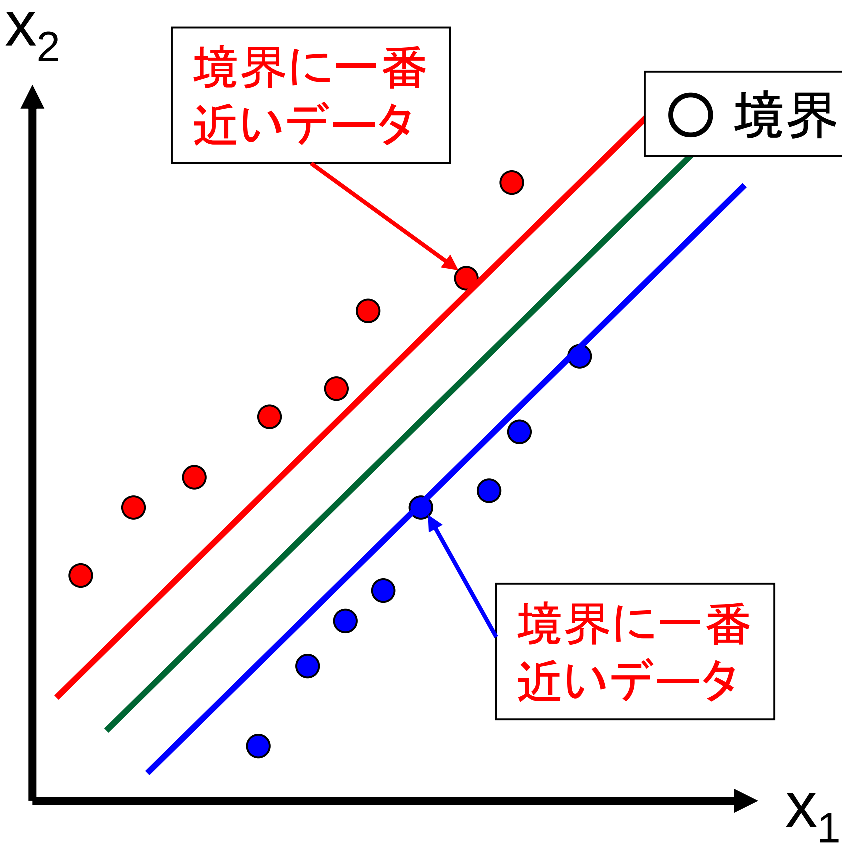
境界に一番近いデータを可能な限り均等に離す

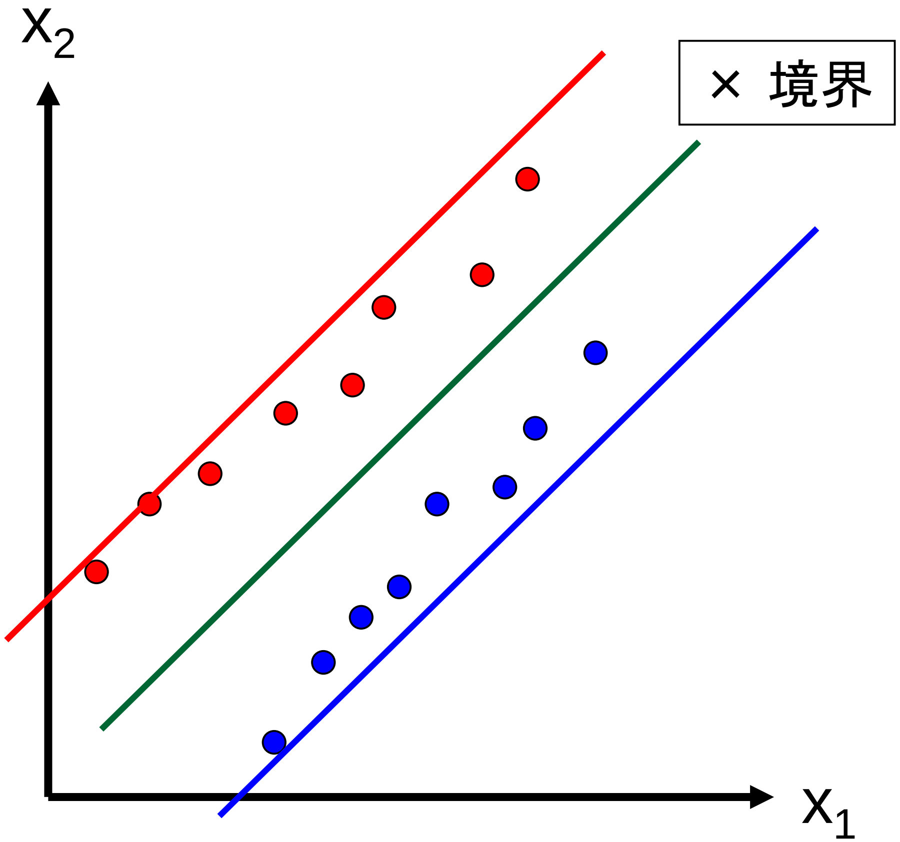




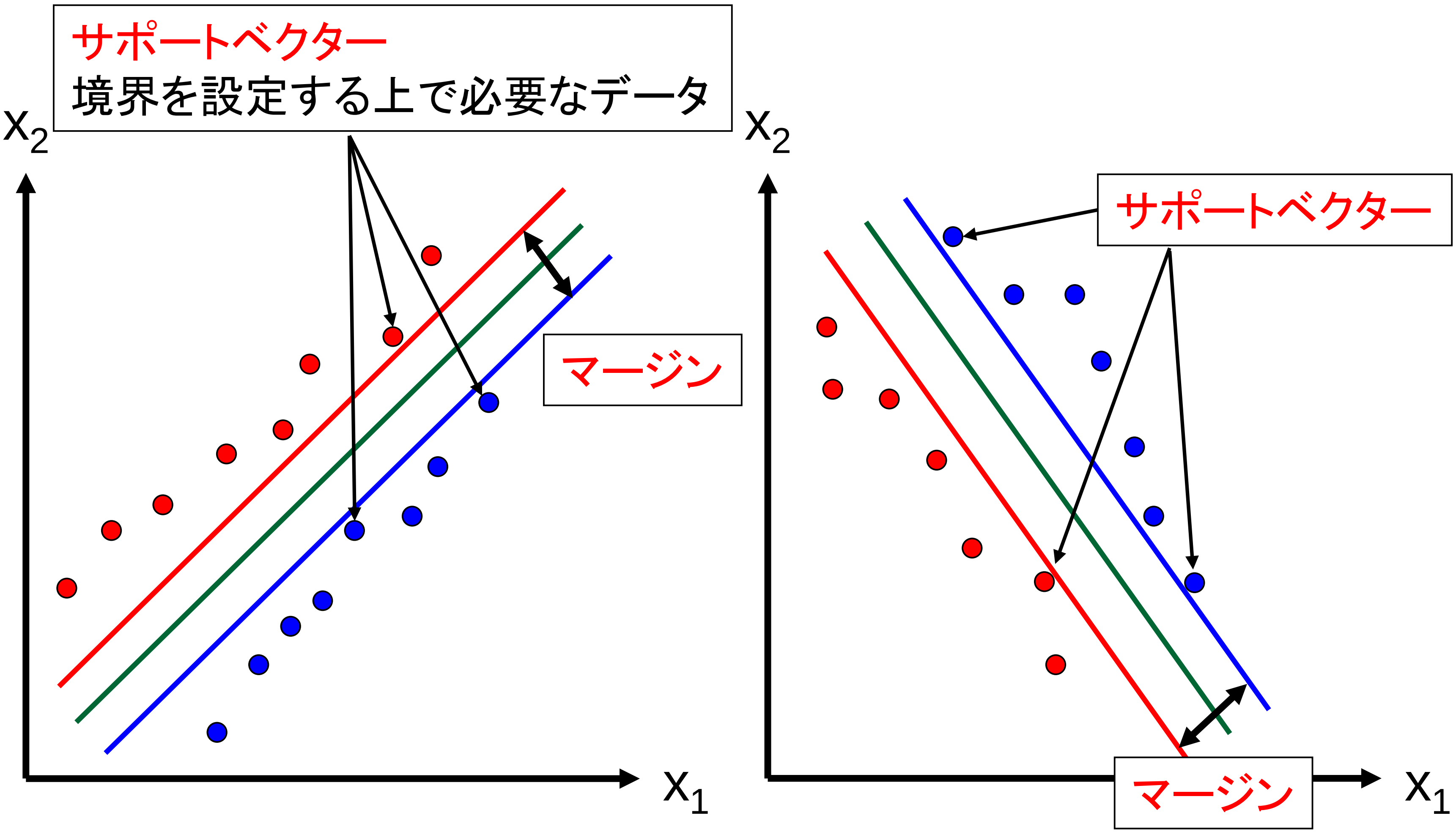
サポートベクターマシンにおける境界の引き方②

境界に一番近いデータを可能な限り均等に離す



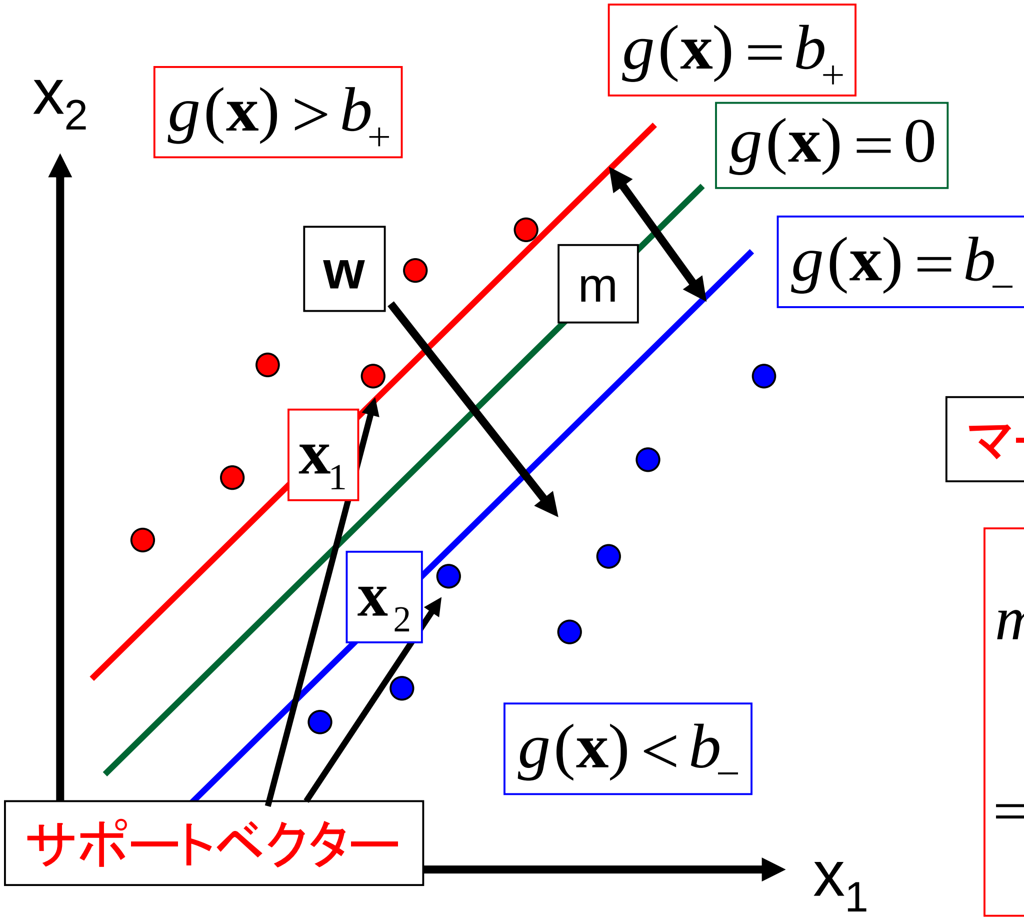


マージン①



境界に近いデータ（サポートベクター）との距離（マージン）が最大になるように，境界線を設定

マージン②



$\mathbf{x}_1$, $\mathbf{x}_2$: サポートベクター

線形識別関数

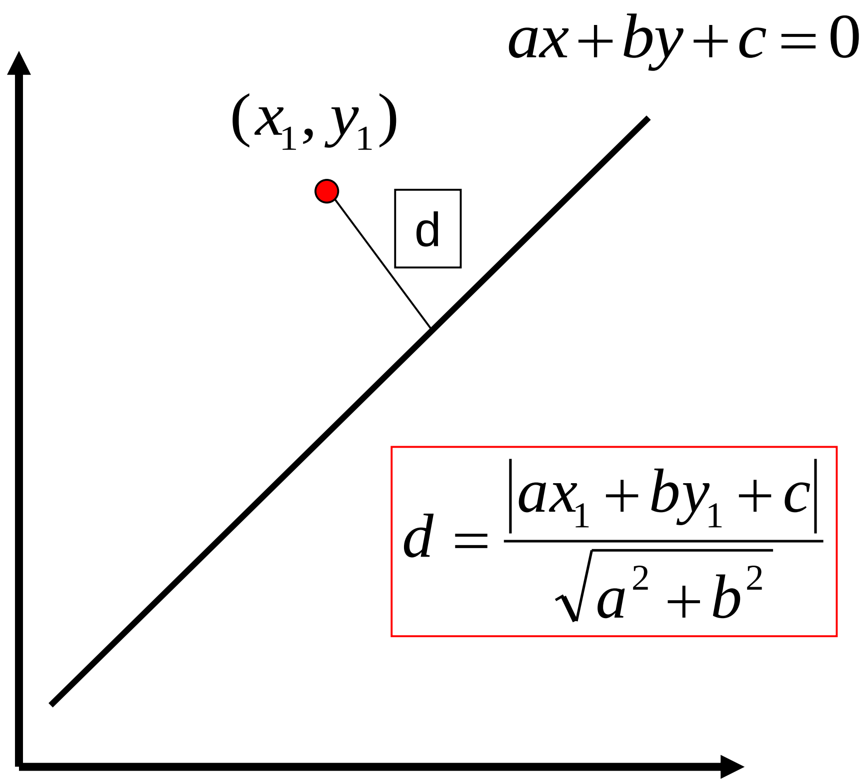
$$g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + b$$

マージン m

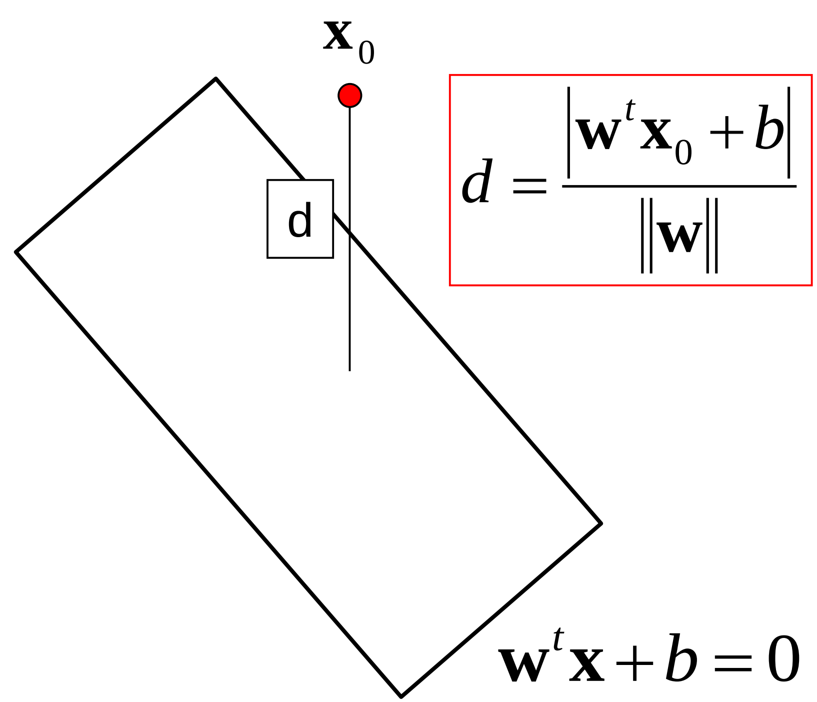
$$m = \frac{|\mathbf{w}^t\mathbf{x}_1 + b| + |\mathbf{w}^t\mathbf{x}_2 + b|}{\|\mathbf{w}\|}$$

$$= \frac{|b_+| + |b_-|}{\|\mathbf{w}\|}$$

垂線の長さ

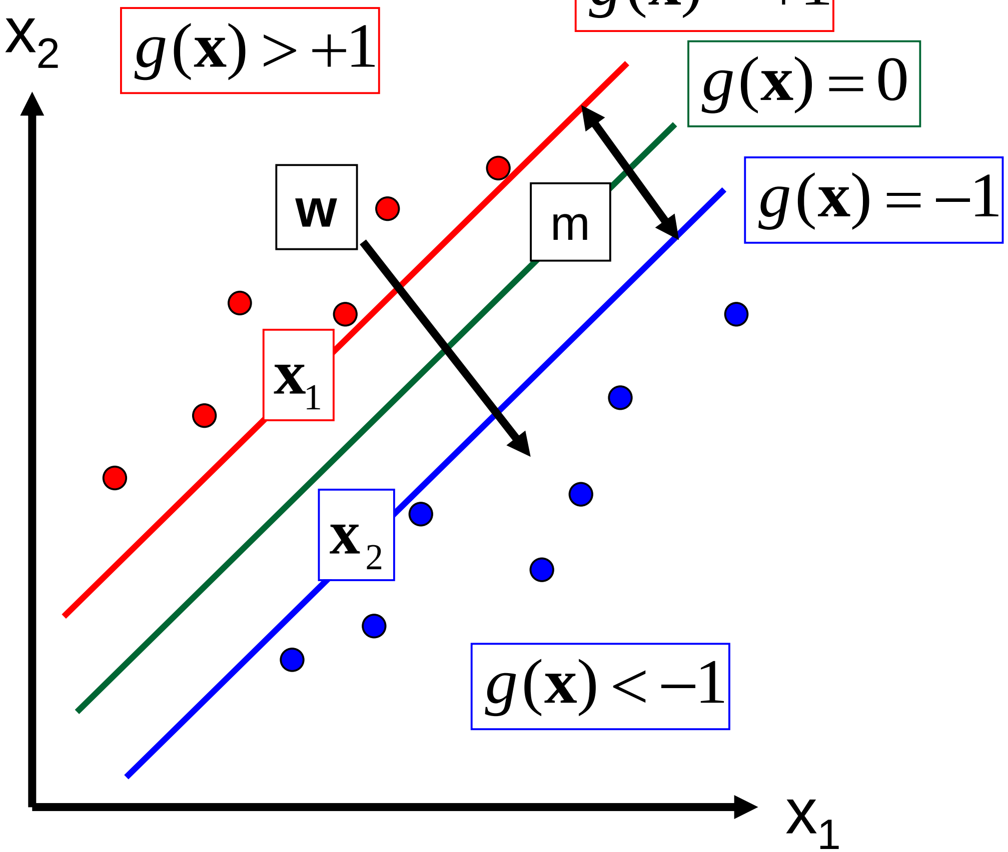


$$d = \frac{|ax_1 + by_1 + c|}{\sqrt{a^2 + b^2}}$$



$$d = \frac{|\mathbf{w}^t\mathbf{x}_0 + b|}{\|\mathbf{w}\|}$$

マージン③



線形識別関数

$$g(\mathbf{x}) = \mathbf{w}^t \mathbf{x} + b$$

$b_+ = +1$　$b_- = -1$

マージン m

$$m = \frac{2}{\|\mathbf{w}\|}$$

定式化(主問題)

- 学習データ: $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$
- 正解ラベル: $y_i \in \{1, -1\}$ (i=1,2,･･･,n)

- 正例: y_i=1のデータ $\rightarrow \mathbf{w}^t\mathbf{x}_i + b \geq 1$
- 負例: y_i=-1のデータ $\rightarrow \mathbf{w}^t\mathbf{x}_i + b \leq -1$

$$y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1$$

主問題

$$\text{Maximize} \quad \frac{2}{\|\mathbf{w}\|}$$

$$\text{subject to} \quad y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1$$

マージン最大化

マージンが最大となるサポートベクターを決める
重み **w** を求める

解法①

Maximize $\dfrac{2}{\|\mathbf{w}\|}$

subject to $y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1$

Minimize $\dfrac{1}{2}\|\mathbf{w}\|^2$

subject to $y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1$

- ラグランジュ未定乗数法

ラグランジュ関数

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^t\mathbf{w} - \sum_i \alpha_i (y_i(\mathbf{w}^t\mathbf{x}_i + b) - 1)$$

① $\alpha_i \geq 0$

② $\alpha_i(y_i(\mathbf{w}^t\mathbf{x}_i + b) - 1) = 0$

③ $y_i(\mathbf{w}^t\mathbf{x}_i + b) - 1 \geq 0$

α i:ラグランジュ定数(非負)

ラグランジュ未定乗数法(不等式制約条件)

- 定式化

$$\min_{\mathbf{x}} \ f(\mathbf{x}) \quad \text{subject to } g_i(\mathbf{x}) \geq 0 \quad \text{for } i = 1, 2, \cdots, n$$

- 次式を満たす $\alpha_i \geq 0$ (i=1,2,・・・,n)が存在
- カルーシュ・クーン・タッカー条件(KKT)

$$L(\mathbf{x}, \boldsymbol{\alpha}) = f(\mathbf{x}) - \sum_{i=1}^{n} \alpha_i g_i(\mathbf{x})$$

$$\left. \frac{\partial L(\mathbf{x}, \boldsymbol{\alpha})}{\partial \mathbf{x}} \right|_{x=x_0} = 0$$

$$\alpha_i g_i(\mathbf{x}) = 0$$

$$g_i(\mathbf{x}) \geq 0$$

ラグランジュ未定乗数法（不等式制約条件）

- 定式化

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{subject to } g_i(\mathbf{x}) \geq 0 \quad \text{for } i = 1,2,\cdots,n$$

- 次式を満たす $\alpha_i \geq 0$（i=1,2,・・・,n）が存在 ➡ ① $\alpha_i \geq 0$
- カルーシュ・クーン・タッカー条件（KKT）

$$L(\mathbf{x},\boldsymbol{\alpha}) = f(\mathbf{x}) - \sum_{i=1}^{n} \alpha_i g_i(\mathbf{x})$$

ラグランジュ関数

$$\left.\frac{\partial L(\mathbf{x},\boldsymbol{\alpha})}{\partial \mathbf{x}}\right|_{x=x_0} = 0$$

$$\alpha_i g_i(\mathbf{x}) = 0$$

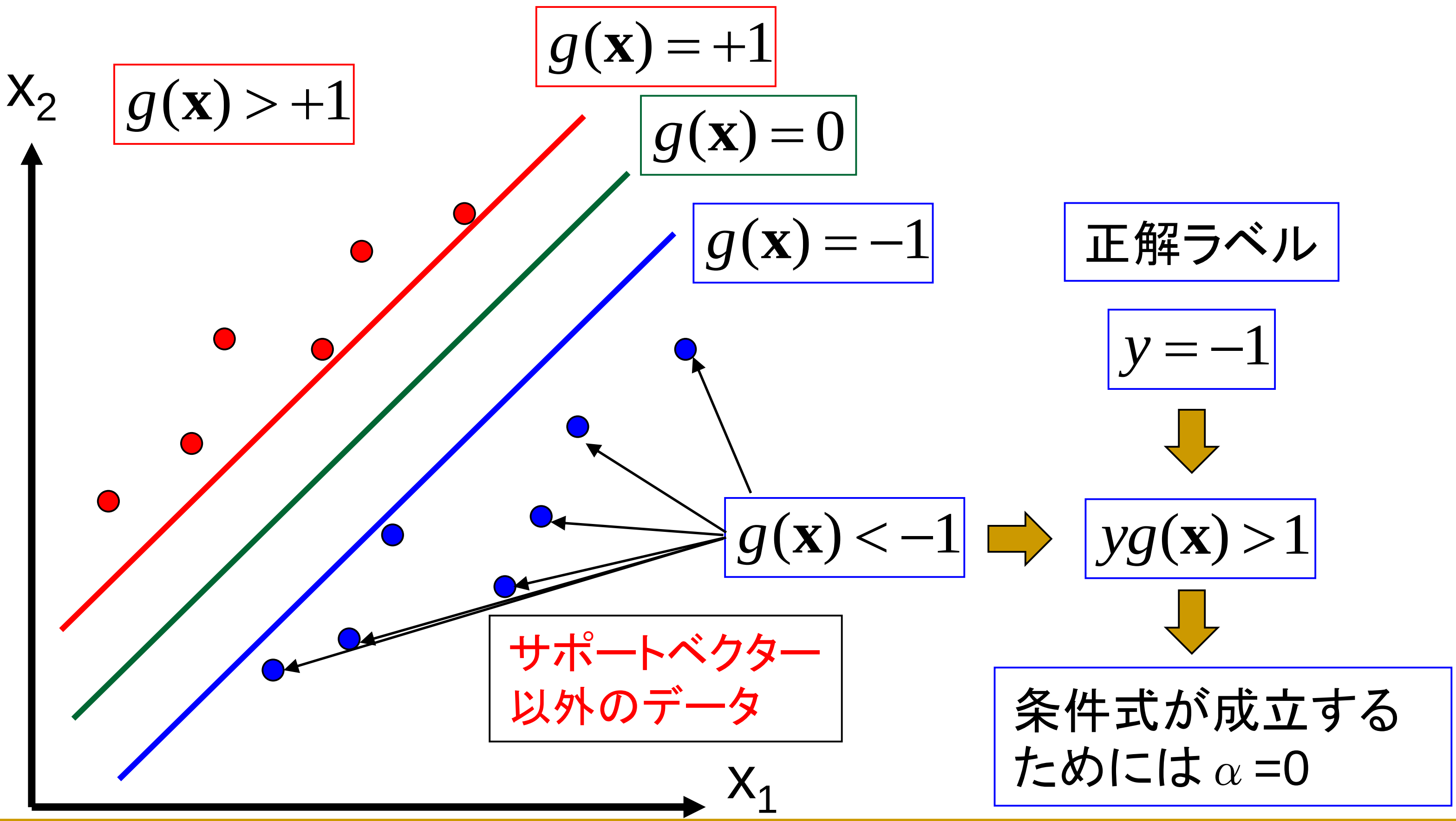
➡ ② $\alpha_i (y_i(\mathbf{w}^t \mathbf{x}_i + b) - 1) = 0$

$$g_i(\mathbf{x}) \geq 0$$

➡ ③ $y_i(\mathbf{w}^t \mathbf{x}_i + b) - 1 \geq 0$

ラグランジュ乗数の意味①

② $\alpha_i(y_i(\mathbf{w}^t\mathbf{x}_i+b)-1)=0$ ⟺ $\alpha_i(y_i g(\mathbf{x}_i)-1)=0$



ラグランジュ乗数の意味②

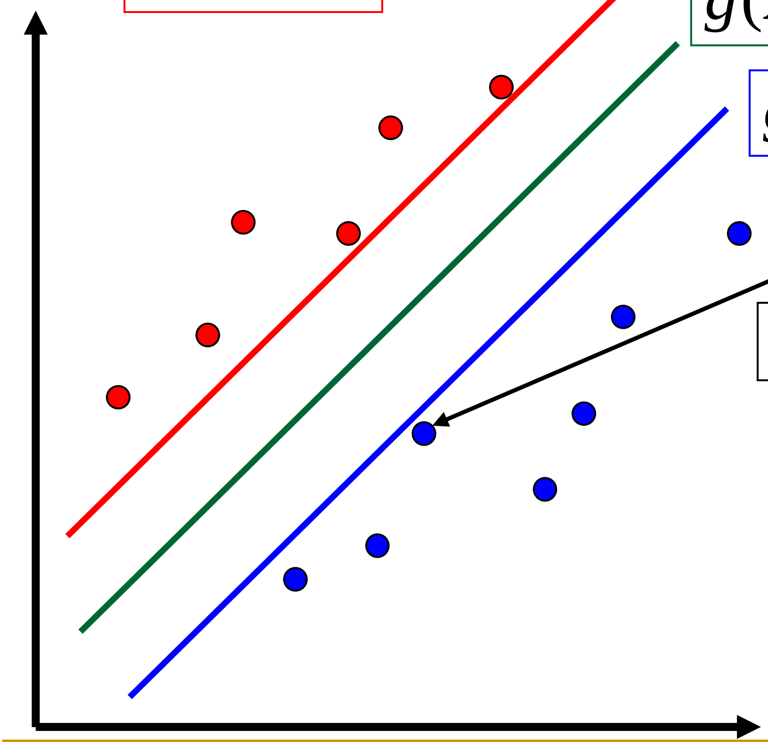
② $\alpha_i(y_i(\mathbf{w}^t\mathbf{x}_i + b) - 1) = 0$ ⇔ $\alpha_i(y_i g(\mathbf{x}_i) - 1) = 0$

$g(\mathbf{x}) > +1$

$g(\mathbf{x}) = +1$

$g(\mathbf{x}) = 0$

$g(\mathbf{x}) = -1$



サポートベクター

正解ラベル

$y = -1$

↓

$g(\mathbf{x}) = -1$ → $yg(\mathbf{x}) = 1$

↓

条件式は成立

↓

α ≧ 0（非負）

↓

α >0となるデータがサポートベクター

次頁

$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$ ⇔ α >0となるデータがサポートベクター

解法②

$$\text{Maximize} \quad \frac{2}{\|\mathbf{w}\|}$$
$$\text{subject to} \quad y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1$$

$$\text{Minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2$$
$$\text{subject to} \quad 1 - y_i(\mathbf{w}^t\mathbf{x}_i + b) \leq 0$$

- ラグランジュ未定乗数法

α $_i$:ラグランジュ定数(非負)

ラグランジュ関数

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^t\mathbf{w} - \sum_i \alpha_i (y_i(\mathbf{w}^t\mathbf{x}_i + b) - 1)$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{w} - \sum_i \alpha_i y_i \mathbf{x}_i = 0 \qquad \frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} = \sum_i \alpha_i y_i = 0$$

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

α $_i$が求まればwも求まる

解法③

ラグランジュ関数

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^t\mathbf{w} - \sum_i \alpha_i \left(y_i (\mathbf{w}^t \mathbf{x}_i + b) - 1 \right)$$

$$= \frac{1}{2}\left(\sum_i \alpha_i y_i \mathbf{x}_i \right)^t \left(\sum_j \alpha_j y_j \mathbf{x}_j \right) - \sum_i \alpha_i \left(y_i (\sum_j \alpha_j y_j \mathbf{x}_j^t \mathbf{x}_i + b) - 1 \right)$$

$$= \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j - \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j - b\sum_i \alpha_i y_i + \sum_i \alpha_i$$

$$= \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j$$

$$\sum_i \alpha_i y_i = 0$$

$$L'(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j$$

α ᵢのみの目的関数

双対問題①

- 双対問題
  - 新しい目的関数の最大化

$$\text{Maximize} \quad L'(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j$$

$$\text{subject to} \quad \alpha_{\mathrm{i}} \geq 0, \sum_i \alpha_i y_i = 0$$

  - 二次計画問題
    - α_iを求める解法
    - SMO（sequential minimal optimization）によって求める

双対問題②

主問題

$$\text{Maximize} \quad \frac{2}{\|\mathbf{w}\|}$$
$$\text{subject to} \quad y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1$$

⇔

双対問題

$$\text{Maximize} \quad L'(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j$$
$$\text{subject to} \quad \alpha_i \geq 0, \sum_i \alpha_i y_i = 0$$

- 主問題
  - **w**を求めた後，α_iを求める
- 双対問題
  - α_iを求めた後，**w**を求める

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

二次計画問題（Quadratic programming）

- SVMの場合の解法
  - SMO（Sequential Minimal Optimization）

α_iの求め方

α_i (i=1,2,・・・,n)を初期化
while KKTを満たさないα_iが存在：
　　KKTを満たさないα_1を選択
　　別のα_2を選択
　　目的関数が改善されるようにα_1とα_2を更新

*従って，SVMで解を求めるには計算時間がかかります

解法④

α_i の多くはゼロ

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

- α $_i$ ＞0のデータがサポートベクター
- S個のサポートベクター：$\mathbf{x}_{tj}$（j=1,2,・・・,S）

$$\boxed{\mathbf{w} = \sum_{j=1}^{S} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}}$$

- 正例のサポートベクター：$\mathbf{x}_+$　$\mathbf{w}^t \mathbf{x}_+ + b = 1$
- 負例のサポートベクター：$\mathbf{x}_-$　$\mathbf{w}^t \mathbf{x}_- + b = -1$

$$\boxed{b = -\frac{1}{2}(\mathbf{w}^t \mathbf{x}_+ + \mathbf{w}^t \mathbf{x}_-)}$$

未知データの予測

① SMOによってα_iを求める

② g(x)を求める

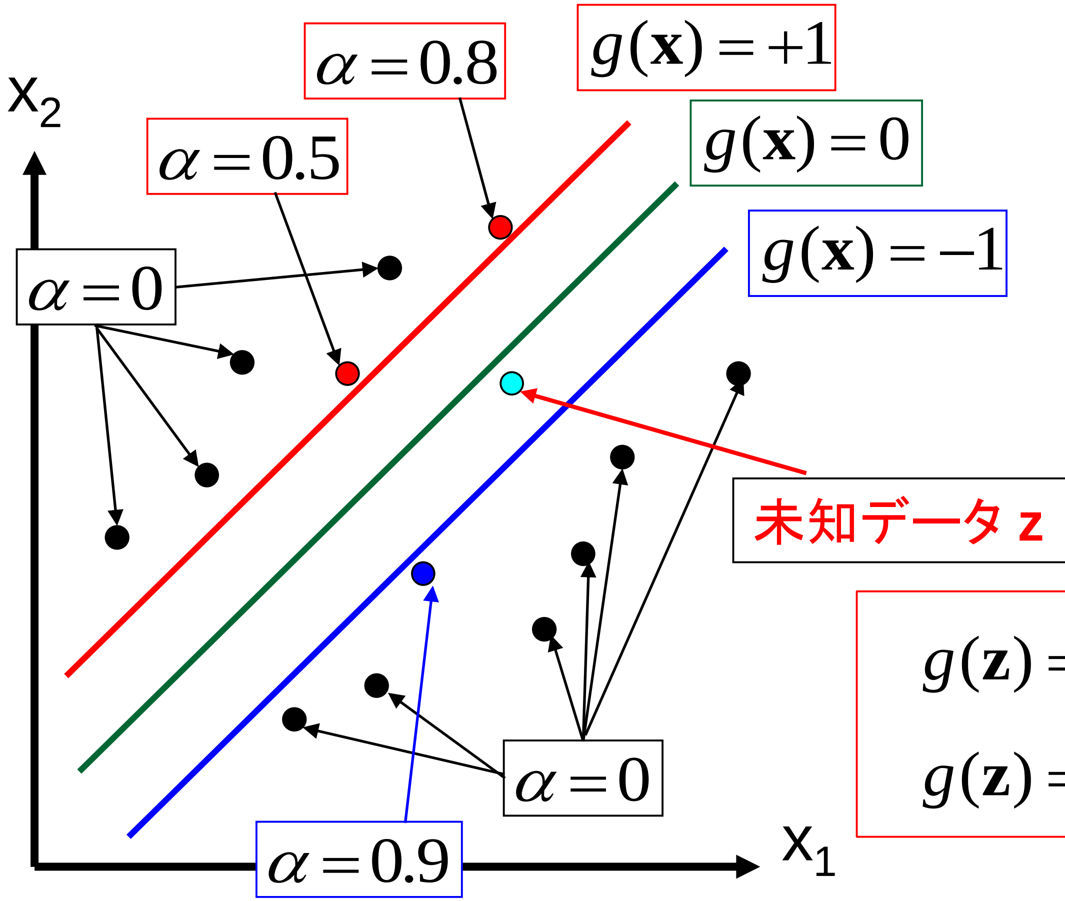
$$\mathbf{w} = \sum_{j=1}^{S} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$$

$$b = -\frac{1}{2}(\mathbf{w}^t \mathbf{x}_+ + \mathbf{w}^t \mathbf{x}_-)$$

③ 未知データzの予測

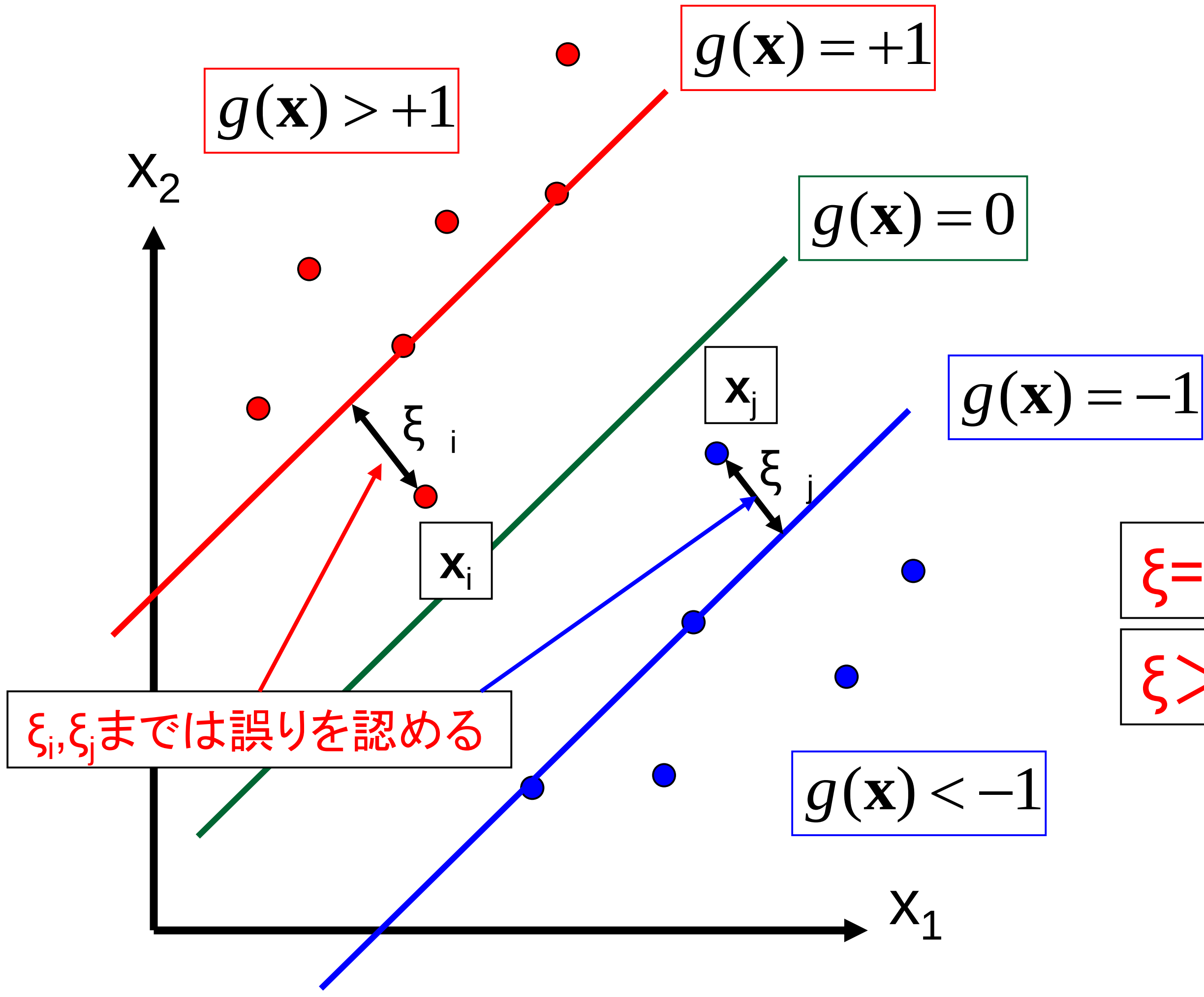
$$g(\mathbf{z}) = \mathbf{w}^t \mathbf{z} + b > 0 \;\rightarrow\; \text{正例}$$

$$g(\mathbf{z}) = \mathbf{w}^t \mathbf{z} + b < 0 \;\rightarrow\; \text{負例}$$



ソフトマージン

条件の緩和（ソフトマージン）



ξ=0 → ハードマージン

ξ>0 → ソフトマージン

ソフトマージンの定式化

- 学習データ: $\mathbf{x}_1$, $\mathbf{x}_2$, ・・・, $\mathbf{x}_n$
- 正解ラベル: $y_i \in \{1, -1\}$ (i=1,2,・・・,n)

- 正例: y_i=1のデータ $\rightarrow \mathbf{w}^t\mathbf{x}_i + b \geq 1 - \xi_i$
- 負例: y_i=-1のデータ $\rightarrow \mathbf{w}^t\mathbf{x}_i + b \leq -1 + \xi_i$

$$y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

$$\text{Minimize} \quad \sum_{\mathrm{i}=1}^{\mathrm{n}} \xi_{\mathrm{i}}$$

「誤り」の二乗和を最小

主問題

$$\text{Minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{\mathrm{i}=1}^{\mathrm{n}} \xi_{\mathrm{i}}$$
$$\text{subject to} \quad y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0$$

C: 定数

ソフトマージンの解法①

$$\text{Minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{\mathrm{i}=1}^{\mathrm{n}} \xi_{\mathrm{i}}$$
$$\text{subject to} \quad y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0$$

- ラグランジュ未定乗数法

ラグランジュ関数

$$L(\mathbf{w}, b, \boldsymbol{\alpha}, \mathbf{v}, \boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^t\mathbf{w} + C\sum_i \xi_i - \sum_i \alpha_i(y_i(\mathbf{w}^t\mathbf{x}_i + b) - 1 + \xi_i) - \sum_i \nu_i \xi_i$$

$$\alpha_i \geq 0$$
$$\alpha_i(y_i(\mathbf{w}^t\mathbf{x}_i + b) - 1 + \xi_i) = 0$$
$$y_i(\mathbf{w}^t\mathbf{x}_i + b) - 1 + \xi_i \geq 0$$

$$\nu_i \geq 0$$
$$\nu_i \xi_i = 0$$
$$\xi \geq 0$$

α_i, ν_i:ラグランジュ定数

ソフトマージンの解法②

$$\text{Minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{\mathrm{i}=1}^{\mathrm{n}} \xi_{\mathrm{i}}$$
$$\text{subject to} \quad y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0$$

- ラグランジュ未定乗数法

ラグランジュ関数

$$L(\mathbf{w}, b, \boldsymbol{\alpha}, \mathbf{v}, \boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^t\mathbf{w} + C\sum_i \xi_i - \sum_i \alpha_i (y_i(\mathbf{w}^t\mathbf{x}_i + b) - 1 + \xi_i) - \sum_i v_i \xi_i$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha}, \mathbf{v}, \boldsymbol{\xi})}{\partial \mathbf{w}} = \mathbf{w} - \sum_i \alpha_i y_i \mathbf{x}_i = 0 \Rightarrow \mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha}, \mathbf{v}, \boldsymbol{\xi})}{\partial b} = \sum_i \alpha_i y_i = 0 \qquad \frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha}, \mathbf{v}, \boldsymbol{\xi})}{\partial \xi_i} = C - \alpha_i - v_i = 0$$

ソフトマージンの解法③

ラグランジュ関数

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

$$L(\mathbf{w}, b, \boldsymbol{\alpha}, \mathbf{v}, \boldsymbol{\xi})$$

$$= \frac{1}{2}\mathbf{w}^t\mathbf{w} + C\sum_i \xi_i - \sum_i \alpha_i (y_i(\mathbf{w}^t\mathbf{x}_i + b) - 1 + \xi_i) - \sum_i \nu_i \xi_i$$

$$= \frac{1}{2}\left(\sum_i \alpha_i y_i \mathbf{x}_i\right)^t \left(\sum_j \alpha_j y_j \mathbf{x}_j\right) + C\sum_i \xi_i - \sum_i \alpha_i \left(y_i (\sum_j \alpha_j y_j \mathbf{x}_j^t \mathbf{x}_i + b) - 1 + \xi_i \right) - \sum_i \nu_i \xi_i$$

$$= \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j + C\sum_i \xi_i - \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j - b\sum_i \alpha_i y_i + \sum_i \alpha_i - \sum_i \alpha_i \xi_i - \sum_i \nu_i \xi_i$$

$$= \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j - b\sum_i \alpha_i y_i + \sum_i (C - \alpha_i - \nu_i)\xi_i$$

$$C - \alpha_i - \nu_i = 0$$

$$\sum_i \alpha_i y_i = 0$$

$$\boxed{L'(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j}$$

α_iのみの目的関数

ソフトマージンにおける双対問題

- 双対問題
  - 新しい目的関数の最大化

$$\text{Maximize} \quad L'(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j$$
$$\text{subject to} \quad \mathrm{C} \geq \alpha_\mathrm{i} \geq 0, \sum_i \alpha_i y_i = 0$$

  - ハードマージンにおける双対問題とほぼ同じ
  - ラグランジュ乗数α $_\mathrm{i}$ に上限値(C)が存在
  - SMOを用いてα $_\mathrm{i}$を求める

$$C - \alpha_i - \nu_i = 0, \nu_i \geq 0 \Rightarrow \mathrm{C} \geq \alpha_\mathrm{i} \geq 0$$

ソフトマージンの解法⑤

α_i の多くはゼロ

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

- $\alpha_i > 0$のデータがサポートベクター
- S個のサポートベクター：$\mathbf{x}_{tj}$（j=1,2,・・・,S）

$$\boxed{\mathbf{w} = \sum_{j=1}^{S} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}}$$

- 正例のサポートベクター：$\mathbf{x}_+$ $\quad \mathbf{w}^t \mathbf{x}_+ + b = 1$
- 負例のサポートベクター：$\mathbf{x}_-$ $\quad \mathbf{w}^t \mathbf{x}_+ + b = -1$

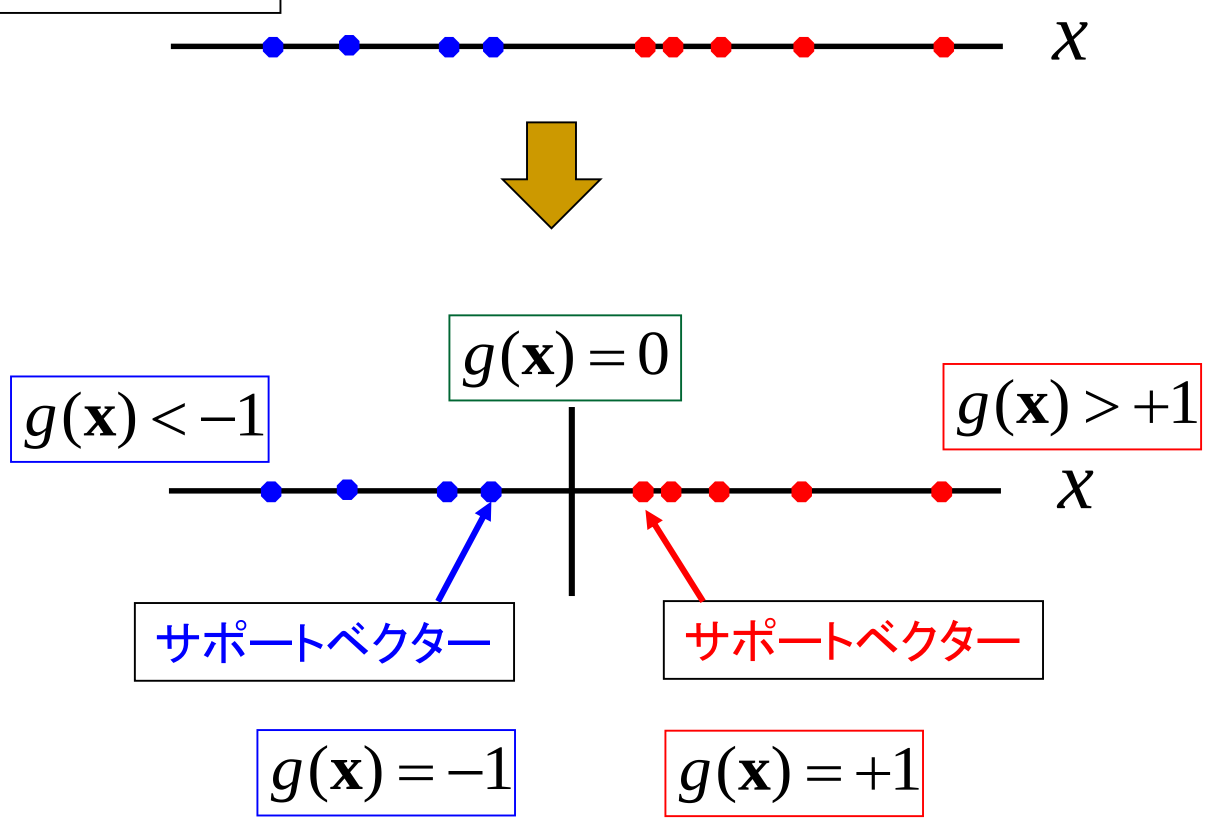
$$\boxed{b = -\frac{1}{2}(\mathbf{w}^t \mathbf{x}_+ + \mathbf{w}^t \mathbf{x}_-)}$$

カーネル法

線形分離不可能な問題への対応①

線形分離可能な場合

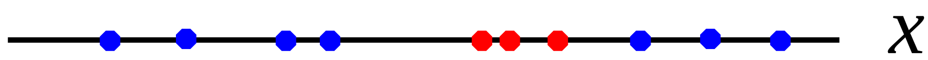
一次元の特徴量



線形分離不可能な問題への対応②

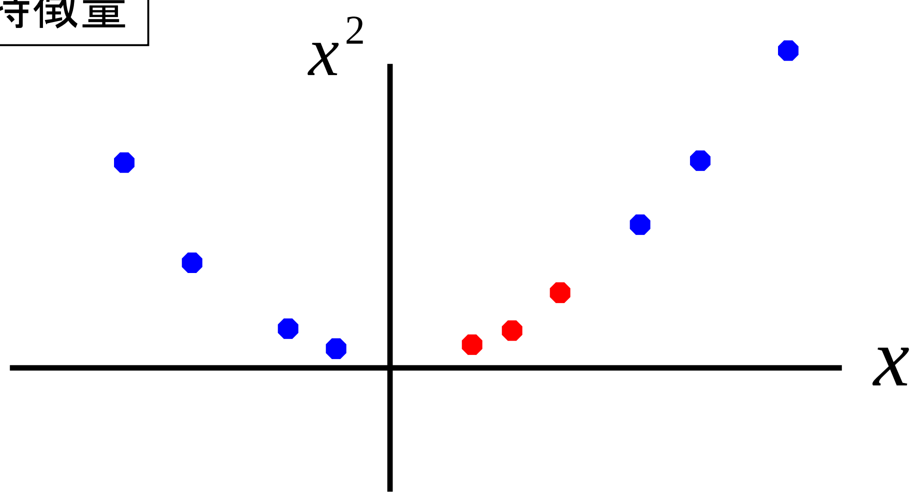
線形分離不可能な場合

一次元の特徴量



(x, x^2) に変換

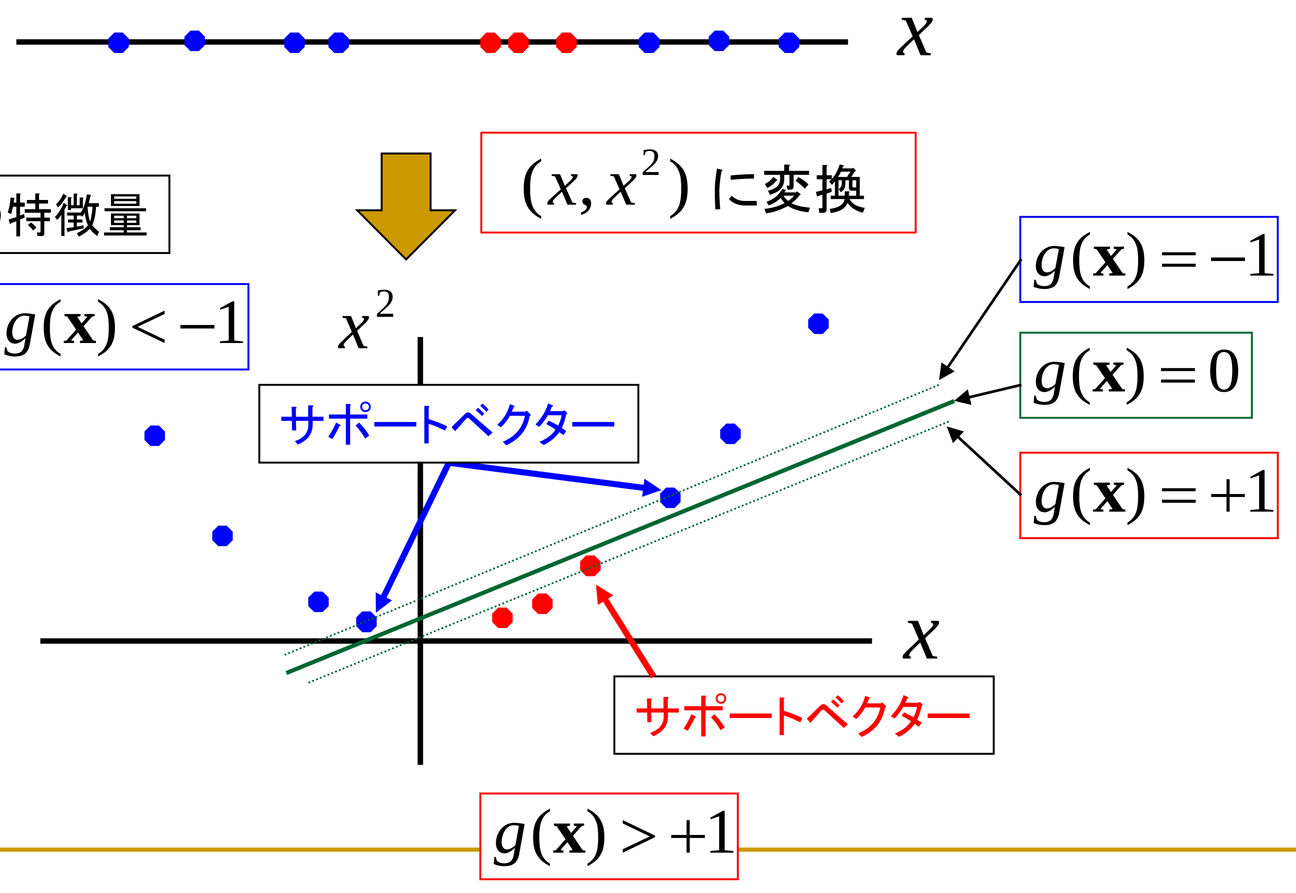
二次元の特徴量



線形分離不可能な問題への対応③

線形分離不可能な場合

一次元の特徴量

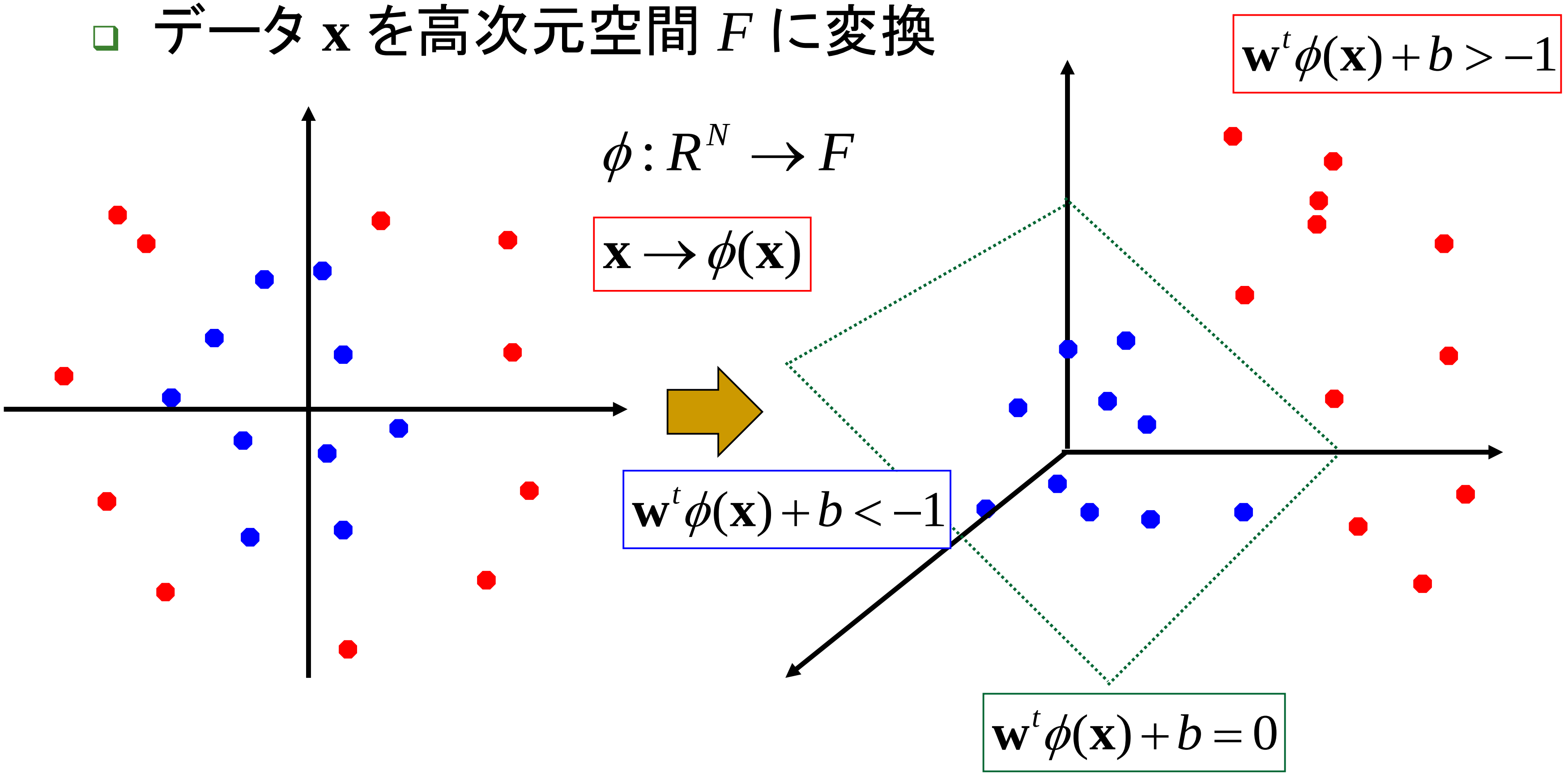


高次元特徴空間への写像

- 非線形写像 $\phi(\mathbf{x})$
  - データ $\mathbf{x}$ を高次元空間 F に変換

$$\phi : R^N \rightarrow F$$

$$\mathbf{x} \rightarrow \phi(\mathbf{x})$$



高次元空間特徴に写像した場合の定式化

- 学習データ: $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$
- 非線形写像 $\phi(\mathbf{x})$
- 正解ラベル: $y_i \in \{1, -1\}$ （i=1,2,・・・,n）

- 正例: $y_i=1$のデータ $\rightarrow$ $\mathbf{w}^t\phi(\mathbf{x}_i)+b \geq 1$
- 負例: $y_i=-1$のデータ $\rightarrow$ $\mathbf{w}^t\phi(\mathbf{x}_i)+b \leq -1$

$\Rightarrow$ $\boxed{y_i(\mathbf{w}^t\phi(\mathbf{x}_i)+b) \geq 1}$

主問題

マージン最大化

$$\text{Maximize} \quad \frac{2}{\|\mathbf{w}\|}$$

$$\text{subject to} \quad y_i(\mathbf{w}^t\phi(\mathbf{x}_i)+b) \geq 1$$

解法: $\mathbf{x} \rightarrow \Phi(\mathbf{x})$と変わったのみで，同じ手順で解く

高次元空間特徴に写像した場合の解法①

- 主問題のラグランジュ関数

$$L(\mathbf{w},b,\boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^t\mathbf{w} - \sum_i \alpha_i (y_i(\mathbf{w}^t\mathbf{x}_i + b) - 1)$$

⇩

$$L(\mathbf{w},b,\boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^t\mathbf{w} - \sum_i \alpha_i (y_i(\mathbf{w}^t\phi(\mathbf{x}_i) + b) - 1)$$

- 双対問題のラグランジュ関数

$$L'(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^t\mathbf{x}_j$$

⇩

$$L'(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \phi(\mathbf{x}_i)^t\phi(\mathbf{x}_j)$$

高次元空間特徴に写像した場合の解法②

- 双対問題

$$\text{Maximize}\quad L'(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \phi(\mathbf{x}_i)^t \phi(\mathbf{x}_j)$$

$$\text{subject to}\quad \alpha_\mathrm{i} \geq 0, \sum_i \alpha_i y_i = 0$$

- α $_\mathrm{i}$をSMOによって求めた後，**w**を求める

$$\mathbf{w} = \sum_i \alpha_i y_i \phi(\mathbf{x}_i)$$

$$b = -\frac{1}{2}(\mathbf{w}^t\phi(\mathbf{x}_+) + \mathbf{w}^t\phi(\mathbf{x}_-))$$

$\mathbf{x}_+$, $\mathbf{x}_-$: サポートベクター

カーネル法

写像Φはどうすればよいのか

- 双対問題のラグランジュ関数

$$L'(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \boxed{\phi(\mathbf{x}_i)^t \phi(\mathbf{x}_j)}$$

高次元特徴空間に変換し，内積を計算するため多くの計算量が必要

$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^t \phi(\mathbf{x}_j)$ となる関数Kを用いる

K:カーネル関数

カーネル関数の例①

$\mathbf{x}^t = (x_1, x_2)$

元のデータ：2次元

$\phi(\mathbf{x}) = (1, x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2)^t$

2次元から6次元への変換

$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^t \mathbf{x}_j)^2$

カーネル関数（多項式カーネル）

$$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^t \mathbf{x}_j)^2$$

元のデータでこの式を計算
→ 元のデータを6次元に変換し，内積を計算したことと同じ
（カーネルトリック）

$$= (1 + x_{i1}x_{j1} + x_{i2}x_{j2})^2$$

$$= \boxed{1 + x_{i1}^2 x_{j1}^2 + 2x_{i1}x_{j1}x_{i2}x_{j2} + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}}$$

$$= (1, x_{i1}^2, \sqrt{2}x_{i1}x_{i2}, x_{i2}^2, \sqrt{2}x_{i1}, \sqrt{2}x_{i2})(1, x_{j1}^2, \sqrt{2}x_{j1}x_{j2}, x_{j2}^2, \sqrt{2}x_{j1}, \sqrt{2}x_{j2})^t$$

$$= \phi(\mathbf{x}_i)^t \phi(\mathbf{x}_j)$$

カーネル関数の例②

$$L'(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \boxed{\phi(\mathbf{x}_i)^t \phi(\mathbf{x}_j)}$$

$$\phi(\mathbf{x}_i)^t \phi(\mathbf{x}_j)$$

高次元空間に変換→内積計算

$$= (1, x_{i1}^2, \sqrt{2}x_{i1}x_{i2}, x_{i2}^2, \sqrt{2}x_{i1}, \sqrt{2}x_{i2})(1, x_{j1}^2, \sqrt{2}x_{j1}x_{j2}, x_{j2}^2, \sqrt{2}x_{j1}, \sqrt{2}x_{j2})^t$$

$$= 1 + x_{i1}^2 x_{j1}^2 + 2x_{i1}x_{j1}x_{i2}x_{j2} + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$$

$$L'(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \boxed{K(\mathbf{x}_i, \mathbf{x}_j)}$$

カーネル関数を用いた場合

$$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^t \mathbf{x}_j)^2$$

$$= (1 + x_{i1}x_{j1} + x_{i2}x_{j2})^2$$

高次元空間に変換しなくてよい
（Φすら知らなくてもよい）

$$= 1 + x_{i1}^2 x_{j1}^2 + 2x_{i1}x_{j1}x_{i2}x_{j2} + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$$

カーネル関数の例③

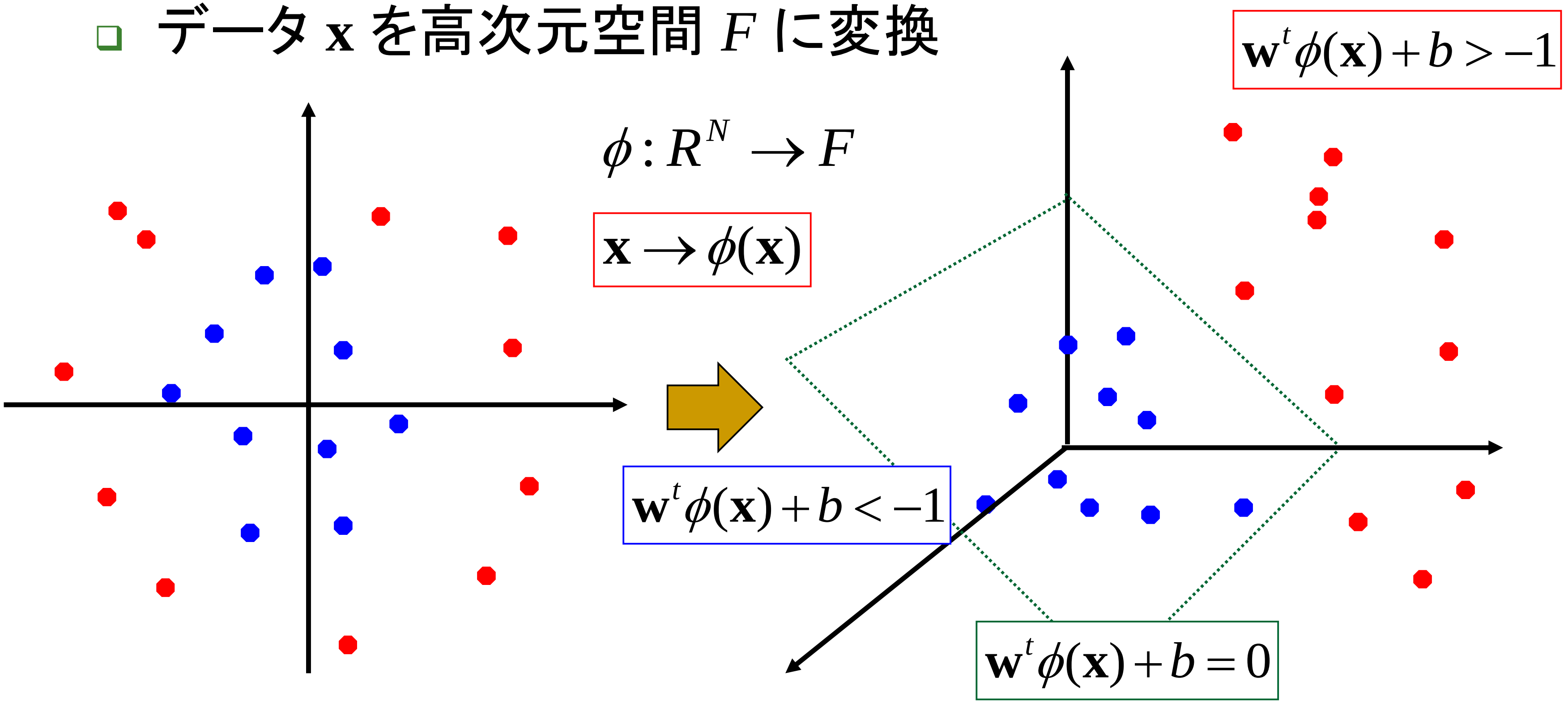
- 線形カーネル $K(\mathbf{x},\mathbf{y}) = \mathbf{x}^t\mathbf{y}$
- 多項式カーネル $K(\mathbf{x},\mathbf{y}) = (1+\mathbf{x}^t\mathbf{y})^p$
- ガウシアンカーネル（RBFカーネル）

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

- シグモイドカーネル $K(\mathbf{x},\mathbf{y}) = \tanh(\alpha\mathbf{x}^t\mathbf{y}+\beta)$

カーネル法の意味

- 非線形写像 $\phi(\mathbf{x})$
  - データ $\mathbf{x}$ を高次元空間 F に変換



  - 非線形写像 $\phi(\mathbf{x})$ を直接用いなくても，カーネル関数により，高次元空間に写像したことになる（カーネルトリック）

未知データの予測①

未知データ **z** の予測

$$g(\mathbf{z}) = \mathbf{w}^t \phi(\mathbf{z}) + b$$

$$= \sum_i \alpha_i y_i \phi(\mathbf{x}_i)^t \phi(\mathbf{z}) + b$$

$$= \sum_i \alpha_i y_i \boxed{K(\mathbf{x}_i, \mathbf{z})} + b$$

$$\mathbf{w} = \sum_i \alpha_i y_i \phi(\mathbf{x}_i)$$

サポートベクターとのカーネル関数を計算

$$g(\mathbf{z}) = \mathbf{w}^t \phi(\mathbf{z}) + b > 0 \rightarrow \text{正例}$$

$$g(\mathbf{z}) = \mathbf{w}^t \phi(\mathbf{z}) + b < 0 \rightarrow \text{負例}$$

未知データの予測②

$\mathbf{x}_+$, $\mathbf{x}_-$: サポートベクター

$$\boxed{\mathbf{w} = \sum_i \alpha_i y_i \phi(\mathbf{x}_i)}$$

$$\mathbf{w}^t \phi(\mathbf{x}_+) + b = 1 \qquad \mathbf{w}^t \phi(\mathbf{x}_-) + b = 1$$

$$\sum_i \alpha_i y_i \phi(\mathbf{x}_i)^t \phi(\mathbf{x}_+) + b = 1 \qquad \sum_i \alpha_i y_i \phi(\mathbf{x}_i)^t \phi(\mathbf{x}_-) + b = 1$$

$$\sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_+) + b = 1 \qquad \sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_-) + b = 1$$

$$\boxed{b = -\frac{1}{2}\left(\sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_+) + \sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_-)\right)}$$

クラス分類問題と回帰問題

クラス分類問題と回帰問題

- クラス分類問題
  - Support Vector Classification（SVC）
- 回帰問題
  - Support Vector Regression（SVR）

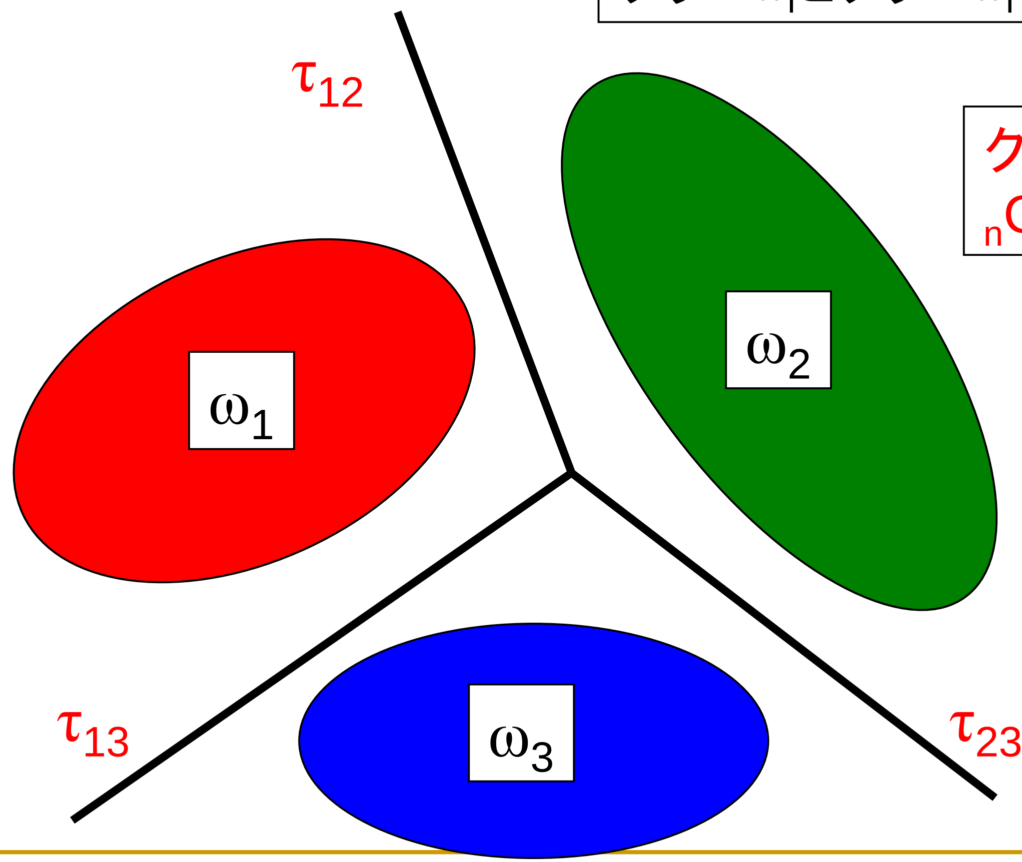
多クラス分類問題への対応

- SVMはニクラス分類問題
- 多クラス分類問題
  - 「1対1」（one-versus-one）による方法
  - 「1対その他」（one-versus-rest）による方法

「1対1による方法」①

- クラス数が3個の場合
  - 特徴ベクトルは2次元

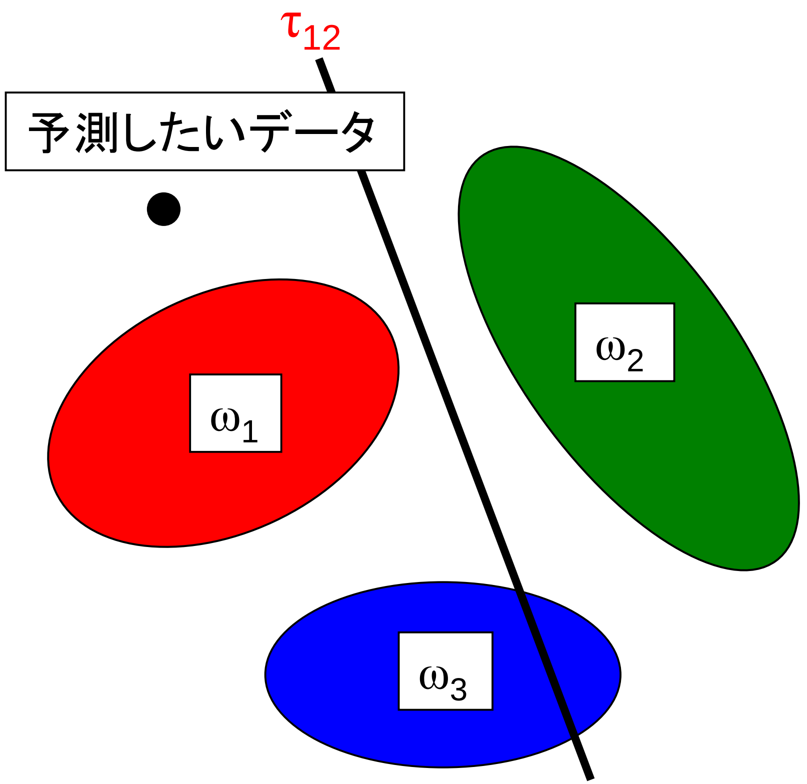
τ_{ij}
クラスω_iとクラスω_jを分ける線形識別関数



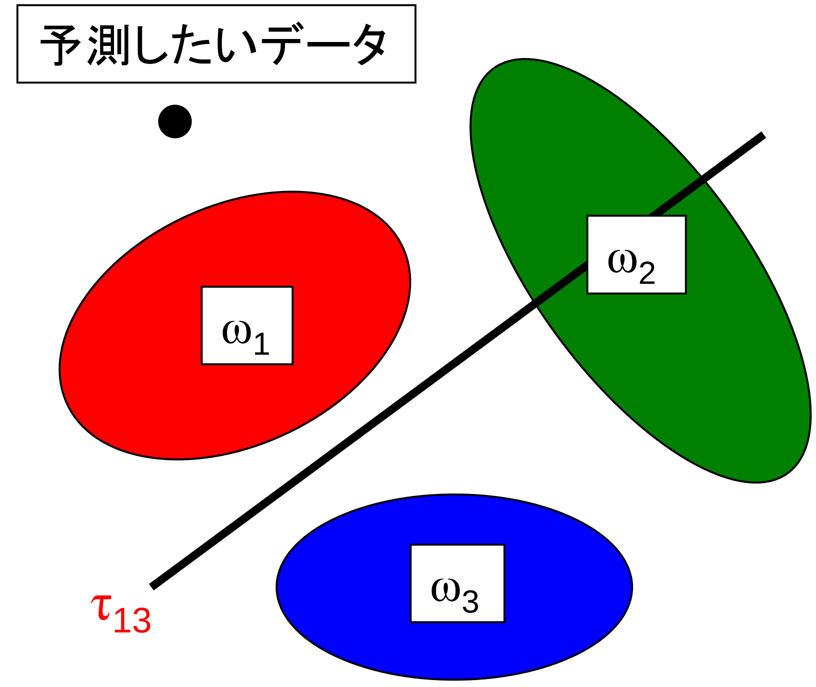
クラス数がn個の場合
${}_nC_2$の線形識別関数

*統計的機械学習のスライド（40ページ）と同じ考え方です

「1対1による方法」②

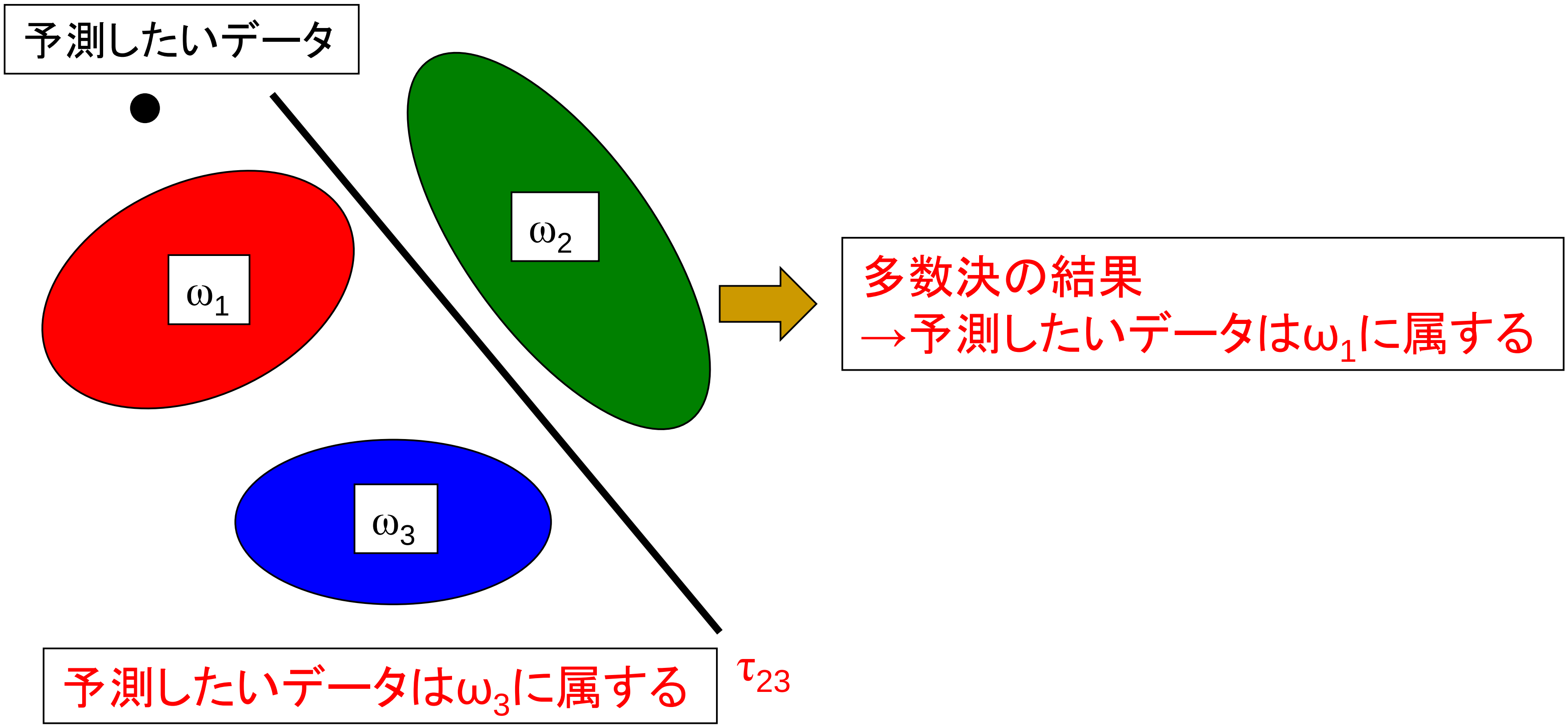


予測したいデータはω_1に属する



予測したいデータはω_1に属する

「1対1による方法」③



予測したいデータ
ω2
ω1
ω3
予測したいデータはω3に属する
τ23
多数決の結果
→予測したいデータはω1に属する

「1対1による方法」④

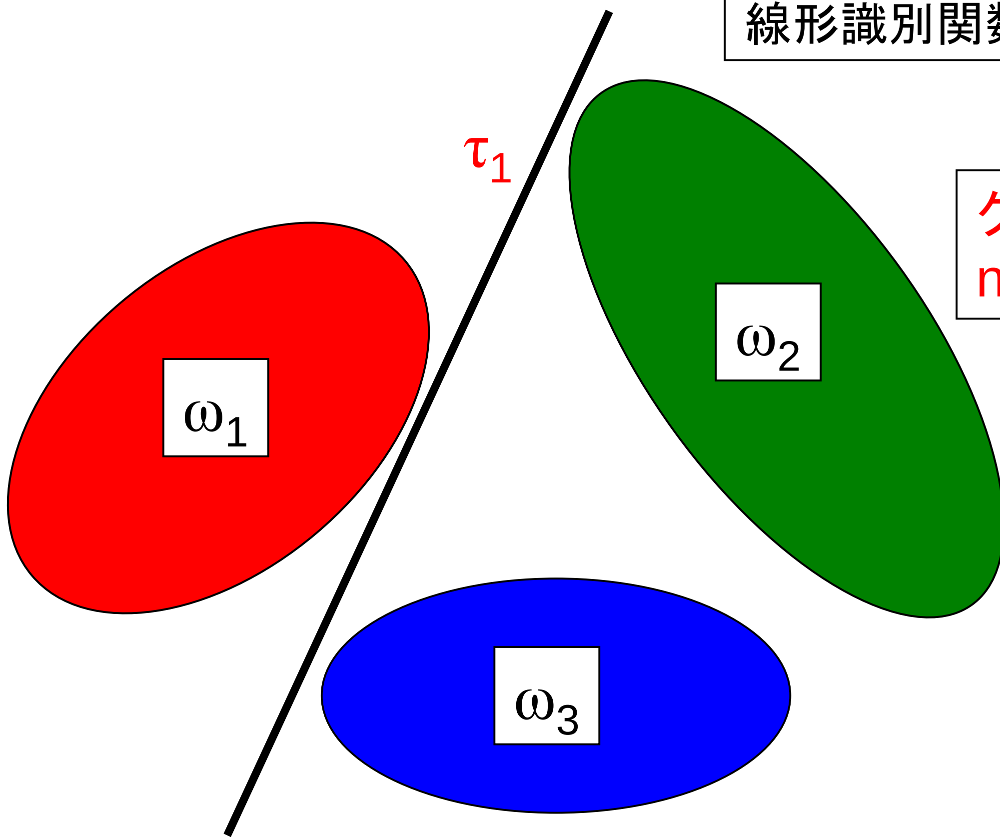
- クラス数がn個の場合
  - 二つのデータを分離する線形識別関数を${}_nC_2$個求める
  - 予測したいデータがどちらに属するか，${}_nC_2$個の線形識別関数において調べる
  - クラス数が多くなった場合，解に矛盾が生じる
  - 多数決によって最終的に解を決める場合もある

「1対その他による方法」①

- クラス数が3個の場合
  - 特徴ベクトルは2次元

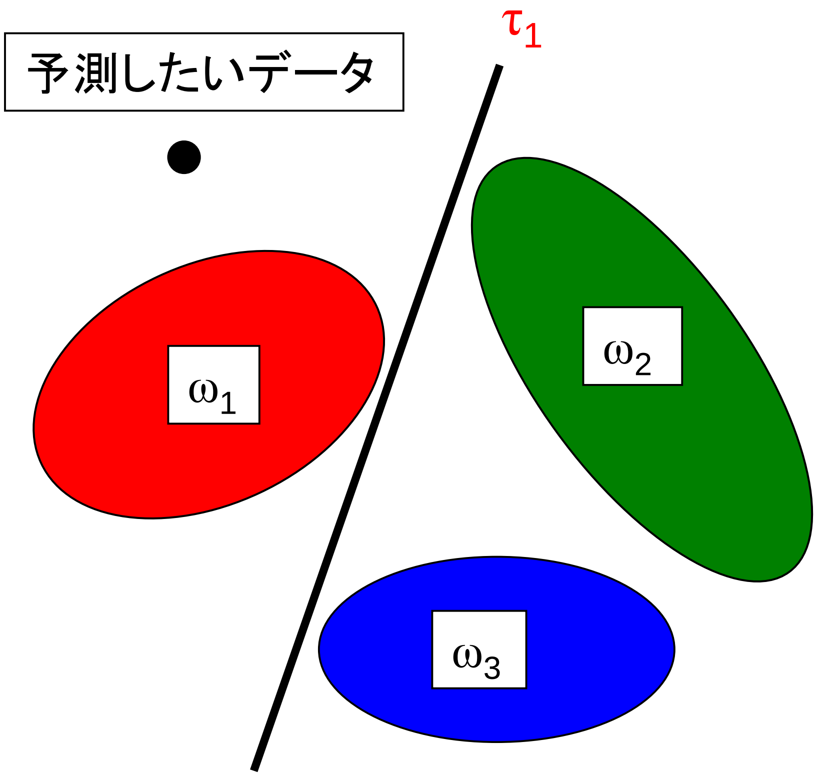
τ_i

クラスω_iとその他のクラスを分ける線形識別関数

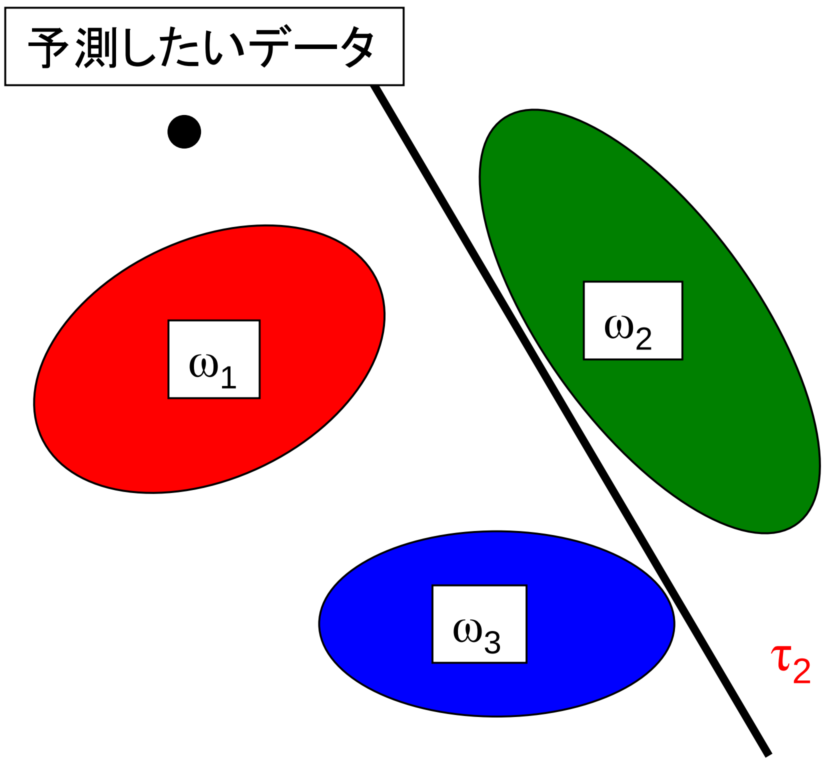


クラス数がn個の場合
n個の線形識別関数

「1対その他による方法」②

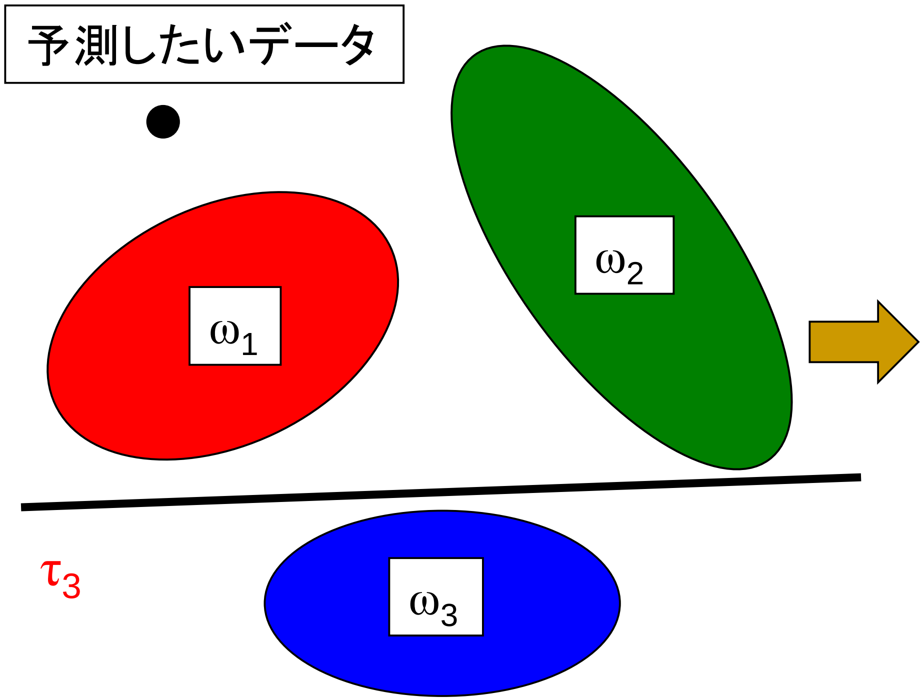


予測したいデータはω_1に属する



予測したいデータはω_1もしくはω_3に属する

「1対その他による方法」③



多数決の結果
→予測したいデータはω_1に属する

予測したいデータはω_1にもしくはω_2に属する

「1対その他による方法」④

- クラス数がn個の場合
  - 任意のクラスとその他のクラスを分離する線形識別関数をn個求める
  - 予測したいデータがどちらに属するか，n個の線形識別関数において調べる
  - クラス数が多くなった場合，解に矛盾が生じる
  - 多数決によって最終的に解を決める場合もある

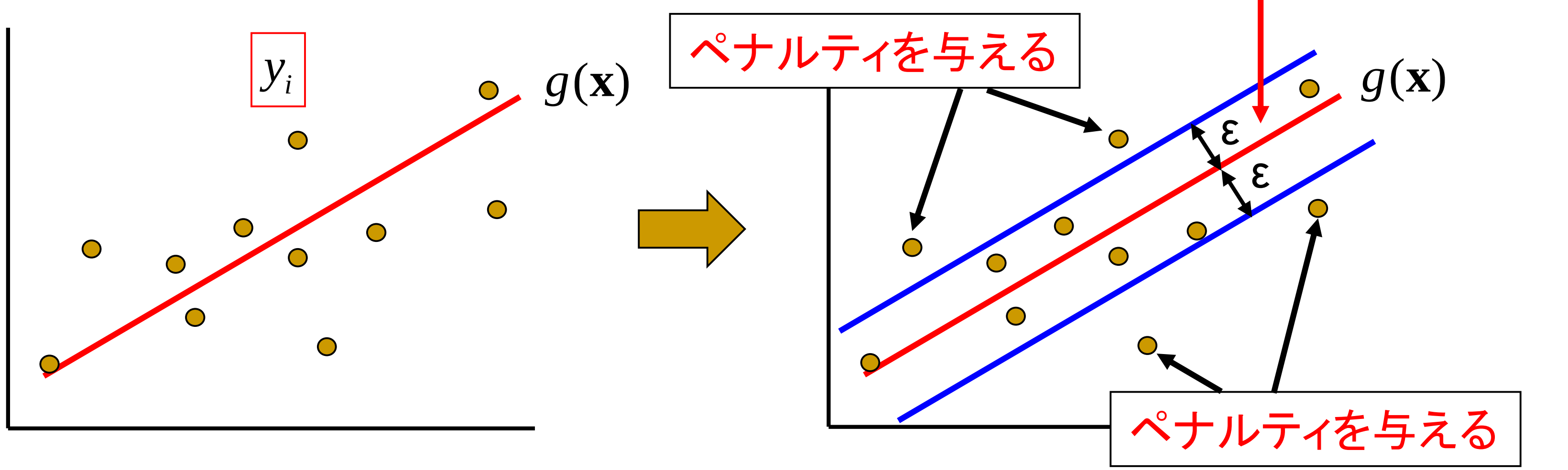
回帰問題への対応

- 入力データ: $\mathbf{x}_1$, $\mathbf{x}_2$, ・・・, $\mathbf{x}_n$
- 正解データ: y_1, y_2, ・・・, y_n

高次元空間への写像：Φ

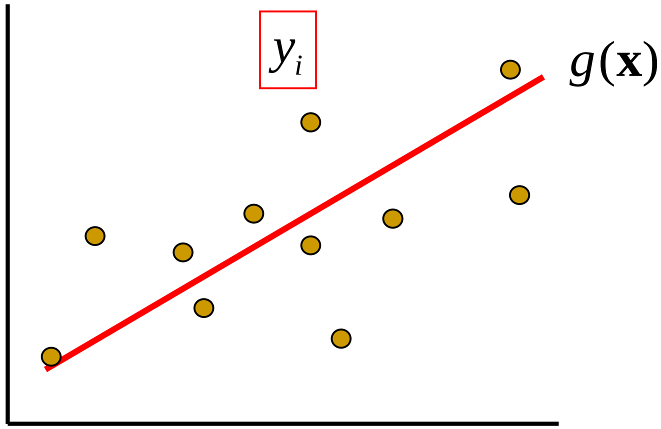
- 回帰モデル　$g(\mathbf{x}) = \mathbf{w}^t \phi(\mathbf{x}) + b$

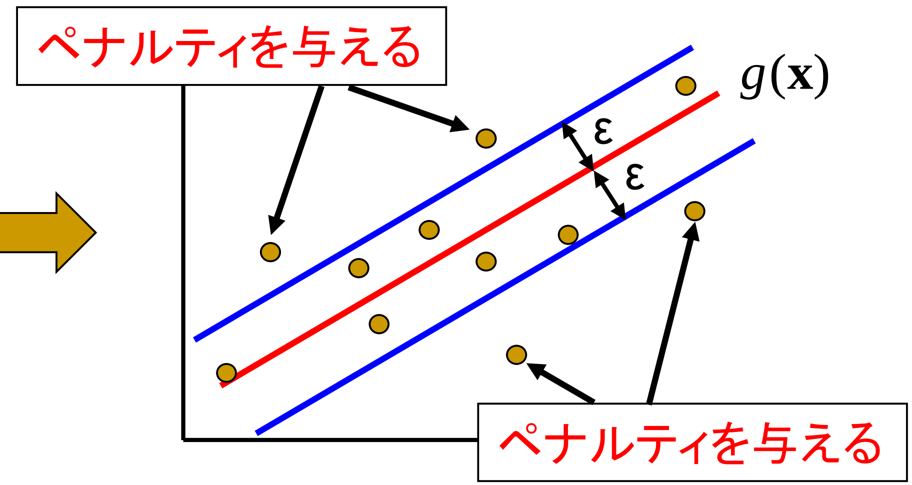
εインセンシティブ
この領域内*のデータには
ペナルティを与えない



*ε-tubeと呼ばれています

εインセンシティブ損失関数①





$$\text{最小化}\ \underbrace{\sum_{i=1}^{n}\left(g(\mathbf{x}_i)-y_i\right)^2}_{\text{誤差項}}+\underbrace{\lambda\|\mathbf{w}\|^2}_{\text{正規化項}}$$

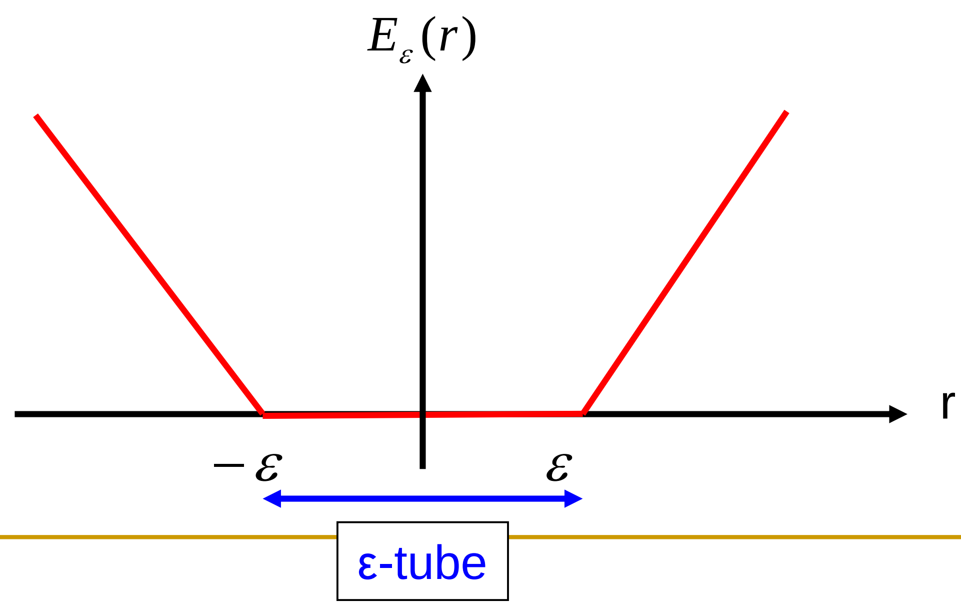
$$\text{最小化}\ C\sum_{i=1}^{n}E_{\varepsilon}\left(g(\mathbf{x}_i)-y_i\right)+\frac{1}{2}\|\mathbf{w}\|^2$$

Eε：εインセンシティブ損失関数
ε-tubeから外れたデータにペナルティを与える

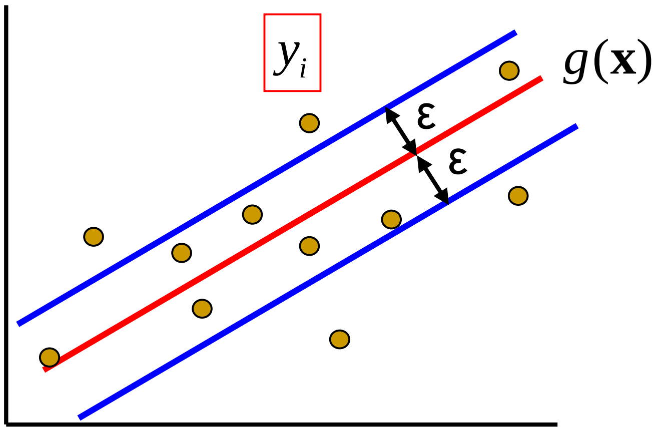
εインセンシティブ損失関数②

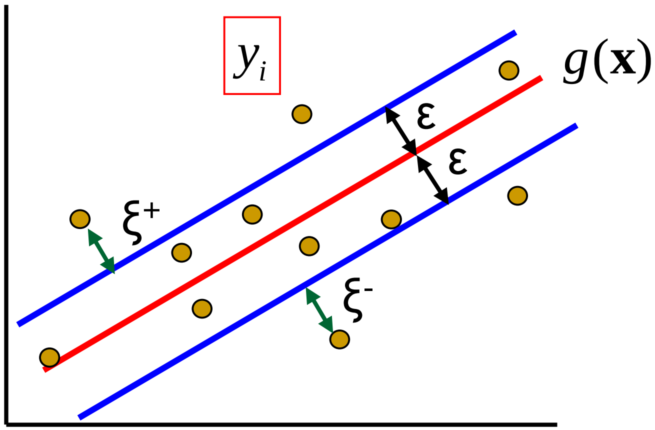
最小化 $C\sum_{i=1}^{n}\boxed{E_{\varepsilon}\left(g(\mathbf{x}_i)-y_i\right)}+\frac{1}{2}\|\mathbf{w}\|^2$

$$\boxed{E_{\varepsilon}(r)=\begin{cases}0 & \text{if } |r|<\varepsilon\\ |r|-\varepsilon & \text{otherwise}\end{cases}}$$



ソフトマージン（スラック変数の導入）





$$g(\mathbf{x}_i) - \varepsilon \le y_i \le g(\mathbf{x}_i) + \varepsilon$$

外れ値にはペナルティを与える

スラック変数ξ+, ξ-

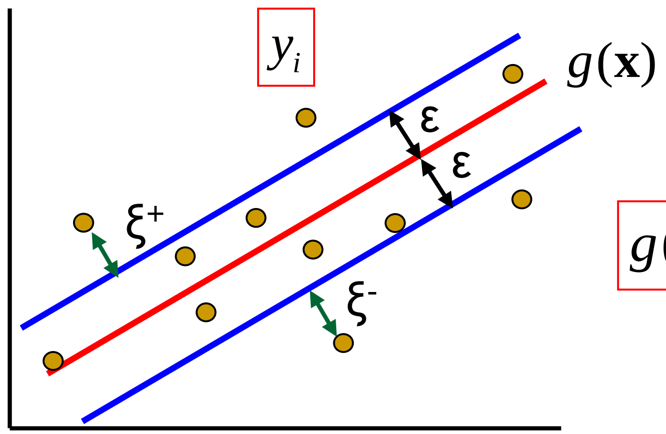
$$\xi_i^+ = \begin{cases} y_i - \varepsilon - g(\mathbf{x}_i) & y_i > g(\mathbf{x}_i) + \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

$$\xi_i^- = \begin{cases} g(\mathbf{x}_i) - y_i - \varepsilon & y_i < g(\mathbf{x}_i) - \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

$$g(\mathbf{x}_i) - \varepsilon - \xi_i^- \le y_i \le g(\mathbf{x}_i) + \varepsilon + \xi_i^+$$

ξ+, ξ-を最小

ソフトマージンによる定式化



$$g(\mathbf{x}_i) - \varepsilon - \xi_i^- \le y_i \le g(\mathbf{x}_i) + \varepsilon + \xi_i^+$$

ξ+, ξ-を最小

主問題

$$\text{minimize} \quad C\sum_{i=1}^{n}(\xi_i^+ + \xi_i^-) + \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{subject to } g(\mathbf{x}_i) - \varepsilon - \xi_i^- \le y_i \le g(\mathbf{x}_i) + \varepsilon + \xi_i^+$$

$$\xi_i^- \ge 0, \xi_i^+ \ge 0$$

SVRの解法①

ラグランジュ関数

$$\lambda(\mathbf{w}, b, \xi_i^+, \xi_i^-, \mu_i^+, \mu_i^-, \alpha_i^+, \alpha_i^-)$$
$$= C\sum_{i=1}^{n}(\xi_i^+ + \xi_i^-) + \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n}(\mu_i^+\xi_i^+ + \mu_i^-\xi_i^-) - \sum_{i=1}^{n}\alpha_i^+(\varepsilon + \xi_i^+ + g(\mathbf{x}_i) - y_i) - \sum_{i=1}^{n}\alpha_i^-(\varepsilon + \xi_i^- - g(\mathbf{x}_i) + y_i)$$

ラグランジュ定数（非負）

$\mu_i^+, \mu_i^-, \alpha_i^+, \alpha_i^-$

KKT条件

$$\mu_i^+ \geq 0 \qquad \mu_i^+\xi_i^+ = 0 \qquad \xi_i^+ \geq 0$$

$$\mu_i^- \geq 0 \qquad \mu_i^-\xi_i^- = 0 \qquad \xi_i^- \geq 0$$

$$\alpha_i^+ \geq 0 \qquad \alpha_i^+(\varepsilon + \xi_i^+ + g(\mathbf{x}_i) - y_i) = 0 \qquad \varepsilon + \xi_i^+ + g(\mathbf{x}_i) - y_i \geq 0$$

$$\alpha_i^- \geq 0 \qquad \alpha_i^-(\varepsilon + \xi_i^- - g(\mathbf{x}_i) + y_i) = 0 \qquad \varepsilon + \xi_i^- - g(\mathbf{x}_i) + y_i \geq 0$$

SVRの解法②

ラグランジュ関数

$$\lambda(\mathbf{w}, b, \xi_i^+, \xi_i^-, \mu_i^+, \mu_i^-, \alpha_i^+, \alpha_i^-)$$
$$= C\sum_{i=1}^{n}(\xi_i^+ + \xi_i^-) + \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n}(\mu_i^+\xi_i^+ + \mu_i^-\xi_i^-) - \sum_{i=1}^{n}\alpha_i^+(\varepsilon + \xi_i^+ + g(\mathbf{x}_i) - y_i) - \sum_{i=1}^{n}\alpha_i^-(\varepsilon + \xi_i^- - g(\mathbf{x}_i) + y_i)$$

$$\frac{\partial\lambda}{\partial\mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{n}(\alpha_i^+ - \alpha_i^-)\phi(\mathbf{x}_i)$$

$$\frac{\partial\lambda}{\partial\xi_i^+} = 0 \Rightarrow \alpha_i^+ + \mu_i^+ = C$$

$$\frac{\partial\lambda}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n}(\alpha_i^+ - a_i^-) = 0$$

$$\frac{\partial\lambda}{\partial\xi_i^-} = 0 \Rightarrow \alpha_i^- + \mu_i^- = C$$

ラグランジュ定数(非負)

$\mu_i^+, \mu_i^-, \alpha_i^+, \alpha_i^-$

双対問題

カーネル関数

最大化

$$\lambda'(\alpha_i^+, \alpha_i^-) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)K(\mathbf{x}_i, \mathbf{x}_j) - \varepsilon\sum_{i=1}^{n}(\alpha_i^+ + \alpha_n^-) + \sum_{i=1}^{n}(\alpha_i^+ - \alpha_i^-)y_i$$

$$\text{subject to } 0 \le \alpha_i^+ \le C, 0 \le a_i^- \le C, \sum_{i=1}^{n}(\alpha_i^+ - a_i^-) = 0$$

SVRの解法③

- 二次計画問題を解き，α $^+$, α $^-$を求める

w，bを求める

$$\mathbf{w} = \sum_{i=1}^{n} (\alpha_i^+ - \alpha_i^-) \phi(\mathbf{x}_i)$$

$$b_+ = y_+ - \mathbf{w}^t \phi(\mathbf{x}_+) - \varepsilon$$

$$b_- = y_+ - \mathbf{w}^t \phi(\mathbf{x}_-) + \varepsilon$$

$$b = \frac{1}{2}(b_+ + b_-)$$

$\mathbf{x}_+$: ε-tubeの上限のデータ
$\mathbf{x}_-$: ε-tubeの下限のデータ

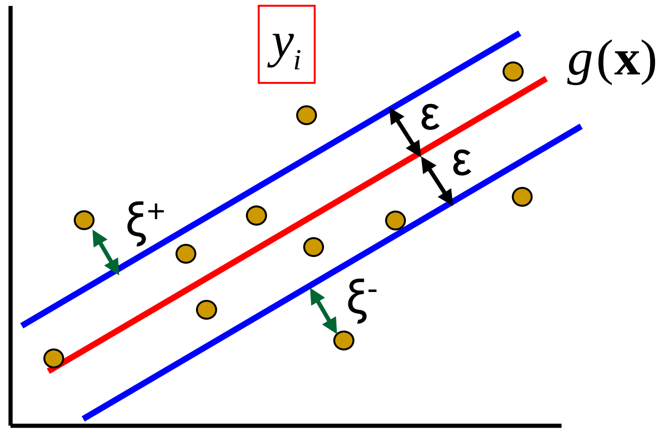
回帰式

$$g(\mathbf{x}_j) = \mathbf{w}^t \phi(\mathbf{x}_j) + b = \sum_{i=1}^{n} (\alpha_i^+ - \alpha_i^-) \phi(\mathbf{x}_i)^t \phi(\mathbf{x}_j) + b$$

$$= \sum_{i=1}^{n} (\alpha_i^+ - \alpha_i^-) K(\mathbf{x}_i, \mathbf{x}_j) + b$$

カーネル関数

サポートベクター



$$\mathbf{w} = \sum_{i=1}^{n} \boxed{(\alpha_i^+ - \alpha_i^-)} \phi(\mathbf{x}_i)$$

0でない場合，wの計算に利用

α＋＞ 0もしくはα－> 0となるデータがサポートベクター

サポートベクターマシンのプログラム

SVM Classification（Iris dataset）

SVM Regression（近似曲線）

Support Vector Classification（SVC）のプログラム（Iris_SVC.py）

- Irisデータセット
  - アヤメの分類問題

用途	クラス分類
データ数	150
特徴量	4
目的変数	3

クラス名	データ数
setosa	50
versicolor	50
virginica	50

SVMによる多クラス分類問題①

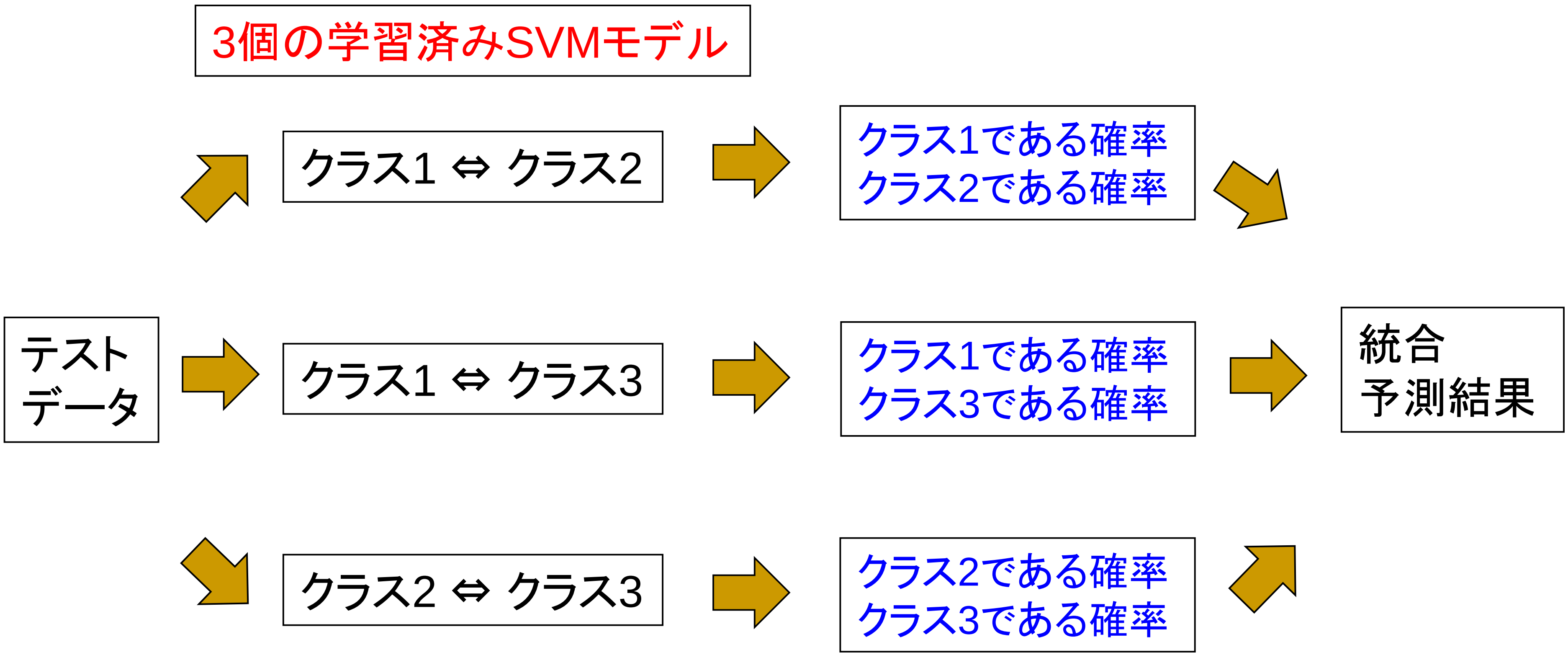
- SVMは二クラス分類問題を対象

- 多クラス分類問題の場合
  - 一対一（one versus one）*
    - 3クラスの場合，3個のモデルを学習
    - ① クラス1とクラス2を分類するSVM
    - ② クラス1とクラス3を分類するSVM
    - ③ クラス2とクラス3を分類するSVM

  - Nクラスの場合→ ${}_{N}C_{2}$個のモデルを学習

*scikit-learnのSVCの場合です．一対多（one versus other）のモデルもあります

SVMによる多クラス分類問題②



Iris_SVC.py

```
import numpy
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score,
confusion_matrix

# データのロード
iris = datasets.load_iris()

# 種類
name = iris.target_names
label = iris.target

# 特徴量
feature_names = iris.feature_names
data = iris.data
```

SVCのために必要

irisデータセットの読み込み

label
目的変数の値（0,1,2）

個数
150

data
特徴量

大きさ
（150,4）

```
# 学習データ, テストデータ
train_data, test_data, train_label, test_label = train_test_split(data, label,
test_size=0.5, random_state=None)

# 多項式カーネル
model = SVC(kernel='poly', C=1, gamma=0.1, probability=True)

# 学習
model.fit(train_data, train_label)


# 予測
predict = model.predict(test_data)
predict_proba = model.predict_proba(test_data)


# サポートベクターの表示
print( " [ Support Vector ]" )
for i in range(3):
    print( i , ":" , model.n_support_[i] )
print( model.support_ )
```

ホールドアウト法

kernel='poly'
多項式カーネル

gamma：
カーネル関数のパラメータ

Cが小さい→ソフトマージン
Cが大きい→ハードマージン

probability=True
→ predict_probaの計算が可能

n_support_
サポートベクターの個数

support_
サポートベクターのデータの番号

```
from sklearn.svm import SVC
model = SVC(kernel='poly', C=1, gamma=0.1, probability=True)
```

kernel='poly'
多項式カーネル

Cが小さい→ソフトマージン
Cが大きい→ハードマージン

gamma：
カーネル関数のパラメータ

probability=True
→ predict_probaの計算が可能

線形カーネル： kernel='linear'
RBFカーネル： kernel='rbf'
多項式： kernel='poly'
シグモイド: kernel='sigmoid'
デフォルトはRBFカーネル

```
model = SVC()
```

```
model = SVC(kernel='rbf', C=1, gamma=0.1, probability=True)
```

カーネル関数のパラメータ

- パラメータ
  - degree, gamma, coef0
  - 多項式カーネル

$$\boxed{K(\mathbf{x},\mathbf{y}) = (\mathrm{coef0} + \mathrm{gamma} \times \mathbf{x}^t\mathbf{y})^{\mathrm{degree}}}$$

  - RBFカーネル

$$\boxed{K(\mathbf{x},\mathbf{y}) = \exp(-gamma \times \|\mathbf{x}-\mathbf{y}\|^2)}$$

  - シグモイドカーネル

$$\boxed{K(\mathbf{x},\mathbf{y}) = \tanh(\mathrm{gamma} \times \mathbf{x}^t\mathbf{y} + \mathrm{coef0})}$$

```
# 予測結果の表示
print( "¥n setosa versicolor virginica -> 予測結果 : 正解ラベル" )
for i in range(20):
    print( " {0:5.3f}  {1:5.3f}  {2:5.3f} -> {3:2d} : {4:2d}".
            format( predict_proba[i][0] , predict_proba[i][1] , predict_proba[i][2] ,
            predict[i] , test_label[i] ) )
```

各クラスの確率

予測結果

正解ラベル

```
print( " [ 予測結果 ]" )
print( classification_report(test_label, predict) )
```

accuracy
precision
recall
F値

```
print( "¥n [ 正解率 ]" )
print( accuracy_score(test_label, predict) )
```

accuracyの表示

```
print( "¥n [ 混同行列 ]" )
print( confusion_matrix(test_label, predict) )
```

混同行列の表示

実行結果①

コマンド プロンプト

```
 [ Support Vector ]
0 : 4
1 : 4
2 : 3
[11 25 30 56 18 24 40 43 49 53 63]

 setosa versicolor virginica -> 予測結果 : 正解ラベル
 0.939  0.040  0.021 -> 0 : 0
 0.019  0.793  0.188 -> 1 : 1
 0.033  0.606  0.361 -> 1 : 2
 0.035  0.515  0.451 -> 2 : 2
 0.212  0.766  0.022 -> 1 : 1
 0.001  0.007  0.992 -> 2 : 2
 0.003  0.043  0.954 -> 2 : 2
 0.030  0.934  0.036 -> 1 : 1
 0.042  0.565  0.392 -> 1 : 2
 0.041  0.932  0.027 -> 1 : 1
 0.942  0.040  0.018 -> 0 : 0
 0.022  0.765  0.213 -> 1 : 1
 0.009  0.104  0.888 -> 2 : 2
 0.026  0.367  0.608 -> 2 : 2
 0.014  0.907  0.079 -> 1 : 1
```

クラス1のサポートベクターの個数

クラス2のサポートベクターの個数

クラス3のサポートベクターの個数

サポートベクターのデータ番号

予測結果

正解ラベル

クラス1の予測確率

クラス2の予測確率

クラス3の予測確率

実行結果②

コマンド プロンプト

```
 [ 予測結果 ]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        21
           1       0.68      1.00      0.81        23
           2       1.00      0.65      0.78        31

    accuracy                           0.85        75
   macro avg       0.89      0.88      0.86        75
weighted avg       0.90      0.85      0.85        75


 [ 正解率 ]
0.8533333333333334

 [ 混同行列 ]
[[21  0  0]
 [ 0 23  0]
 [ 0 11 20]]
```

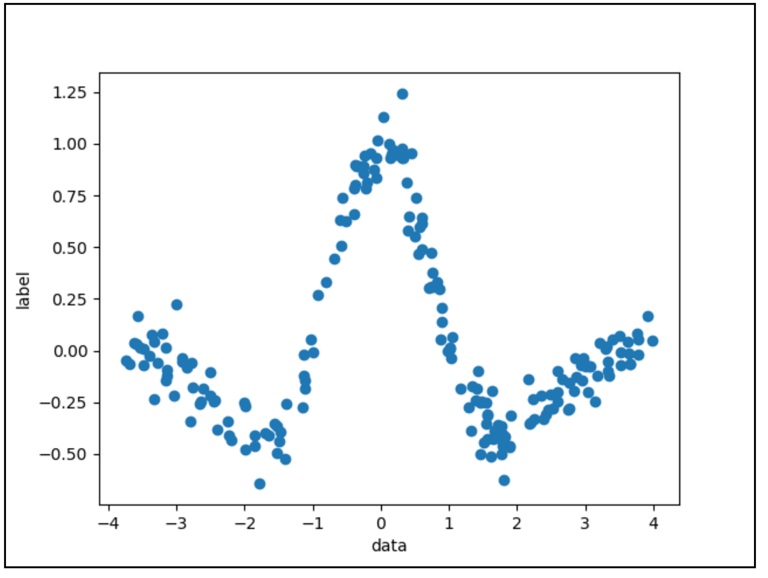
accuracy
precision
recall
F値

accuracyの表示

混同行列の表示

SVM Regression（SVR）のプログラム（SVR.py）

- メキシカンハット関数の学習



$$f(x) = (1 - x^2) \times \exp(-0.5x^2)$$

SVR.py

```
import numpy as np
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.svm import SVR
from sklearn.metrics import classification_report, accuracy_score,
confusion_matrix
import matplotlib.pyplot as plt


# メキシカンハット関数
data = np.sort(np.random.uniform(-4, 4, 200))
label = ( 1 - data*data ) * np.exp( -0.5 * data * data ) +
            0.1*np.random.normal(size=len(data))


data = data.reshape( (len(data),1) )
```

SVRのために必要

np.random.uniform(L, H, N)
L以上，H未満の一様乱数をN個生成

$$f(x) = (1 - x^2) \times \exp(-0.5x^2)$$

np.random.normal(N)
N個の標準正規乱数を生成

（200，1）の行列に変形

```
# RBFカーネル
model = SVR(kernel='rbf', C=10000, gamma=0.1, epsilon=0.1)
```

epsilon：
ε-tubeの長さ

kernel='rbf'
多項式カーネル

Cが小さい→ソフトマージン
Cが大きい→ハードマージン

gamma：
カーネル関数の係数

```
# 学習
model.fit(train_data, train_label)

# 予測（テストデータ）
predict = model.predict(test_data)

# R2を求める
train_score = model.score(train_data, train_label)
test_score = model.score(test_data, test_label)

print( "¥n [ R2 ]" )
print( " 学習データ　: {0:7.5f}".format( train_score ) )
print( " テストデータ: {0:7.5f}".format( test_score ) )
```

散布図の描画

```
fig = plt.figure()
plt.scatter( test_data , predict , c="red" , label="predict" )
plt.scatter( test_data , test_label , c="blue" , label="data" )
plt.xlabel("data")
plt.ylabel("label")
plt.legend(loc='upper left')
fig.savefig("result.png")
```

予測結果

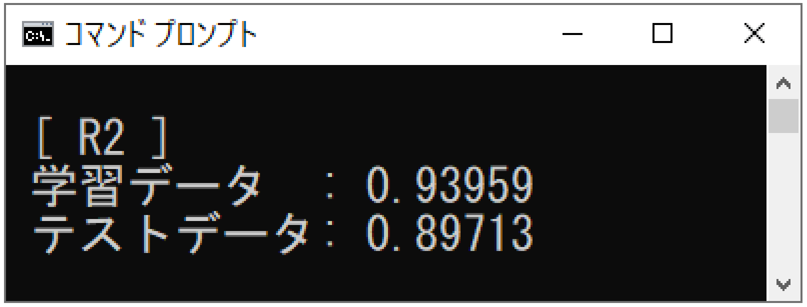
実際のデータ

X軸，Y軸の名前

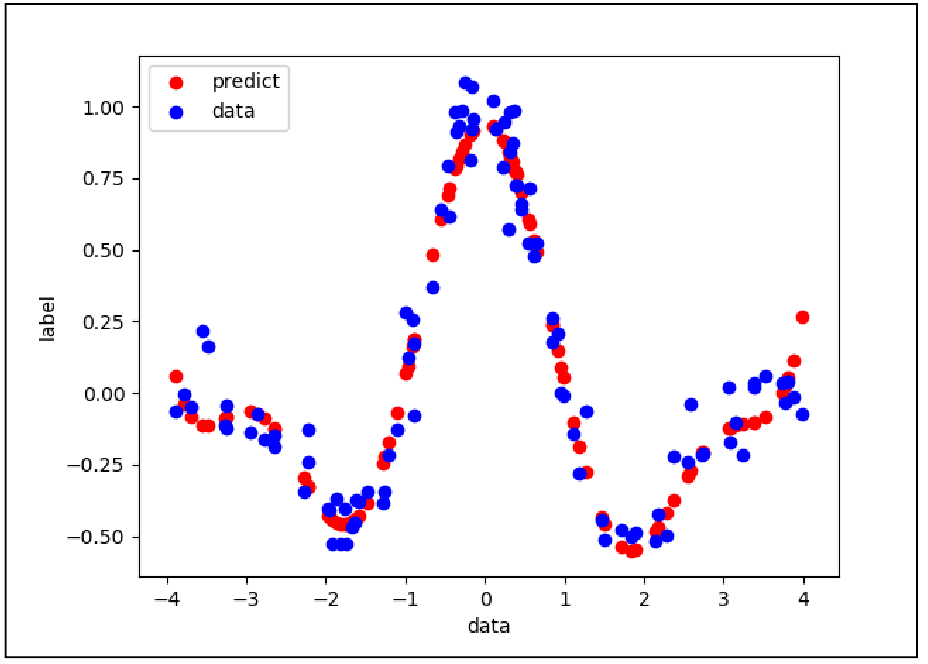
ラベル位置の設定

ファイルに保存

実行結果

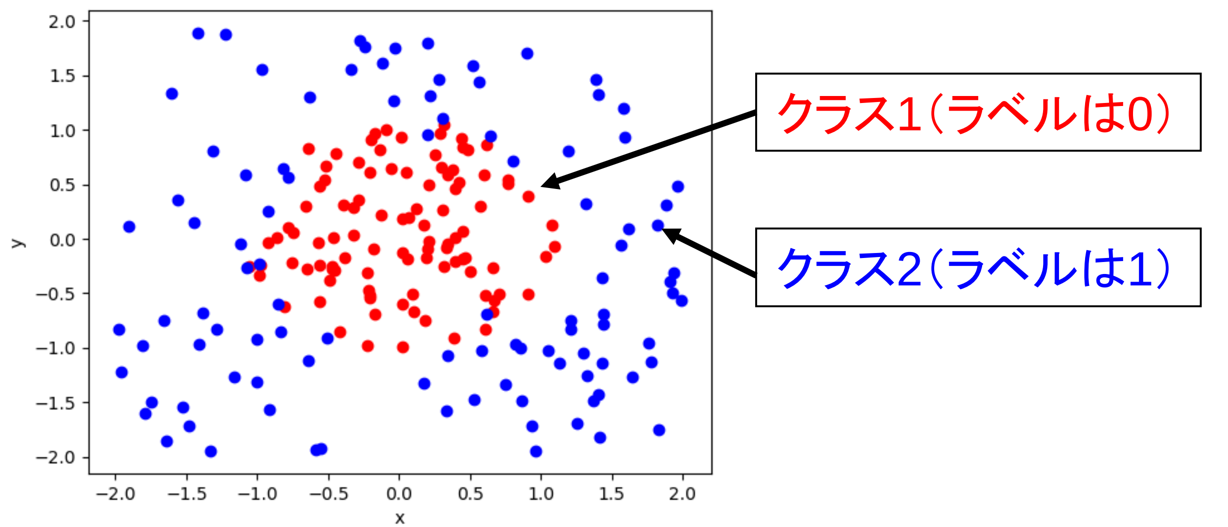


相関係数の二乗



練習問題

- 二値分類
  - SVC-exercise.py を実行して下さい
  - 下図(data.png)のように，クラス1(赤点)，クラス2(青点)のデータ(二次元データ)を100個ずつ生成します．



*宿題とします．12/23(月)13時までにkeio.jpにプログラム(python)とワープロに実行画面を貼り付けて提出して下さい．

練習問題

- データは次の配列に格納されています.
  - クラス1のデータ data[0:100] ラベル label[0:100]
  - クラス2のデータ data[100:]   ラベル label[100:]
- データをホールドアウト法により，学習データ，テストデータに半分ずつに分けなさい.

- SVMによって学習データを学習し，テストデータのラベルを予測しなさい.
- カーネルについては，RBFカーネル，多項式カーネルを試し，カーネルの違いによって分類できる，分類できないことを確認しなさい.

参考文献

- 加藤直樹他：データマイニングとその応用，朝倉書店，2008
- 大北剛訳：サポートベクターマシン入門，共立出版，2005
- 小野田崇：サポートベクターマシン，オーム社，2007
- 平井有三：はじめてのパターン認識，森北出版，2012
- 後藤正幸：入門 パターン認識と機械学習，コロナ社，2014

参考文献

- SVC
- https://scikit-learn.org/stable/modules/generated/
sklearn.svm.SVC.html

- SVR
- https://scikit-learn.org/stable/modules/generated/
sklearn.svm.SVR.html