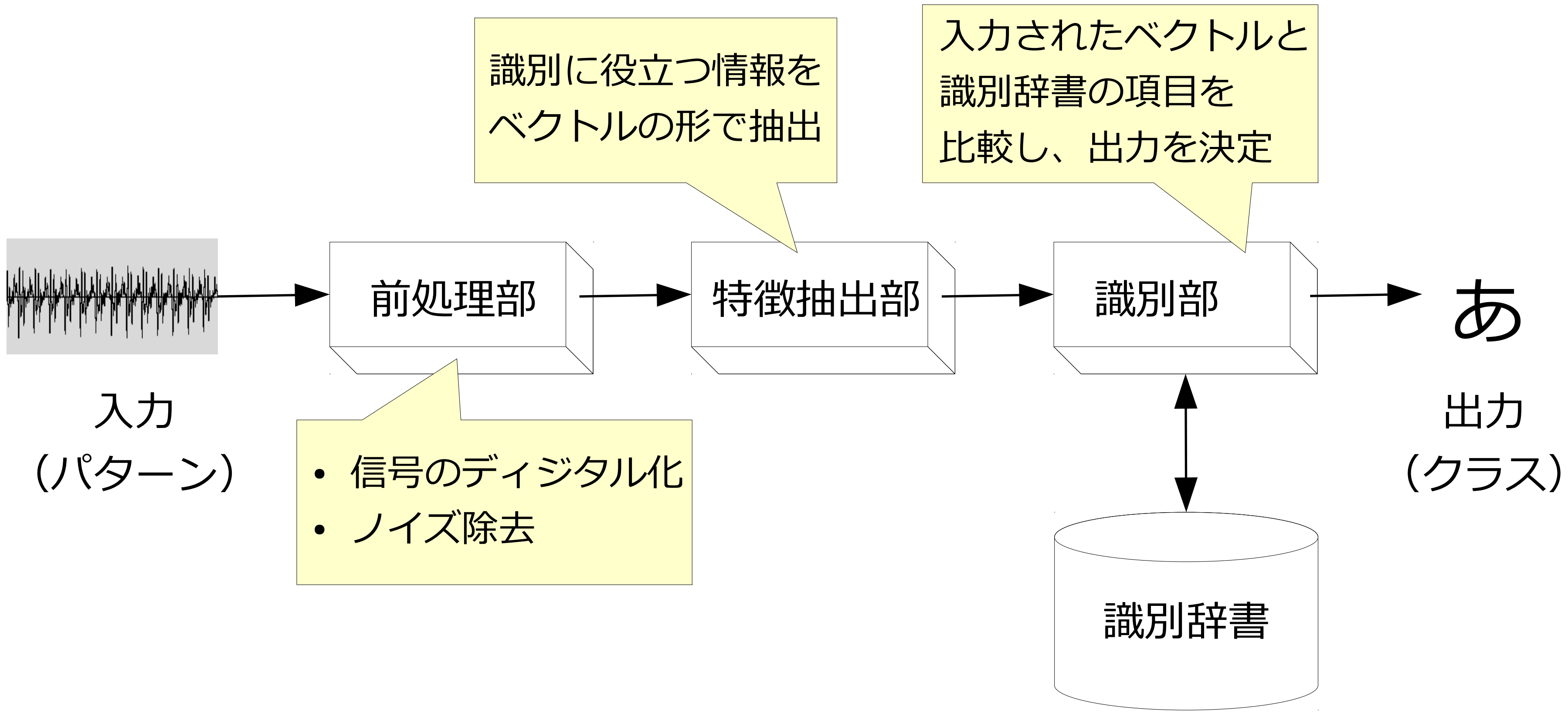
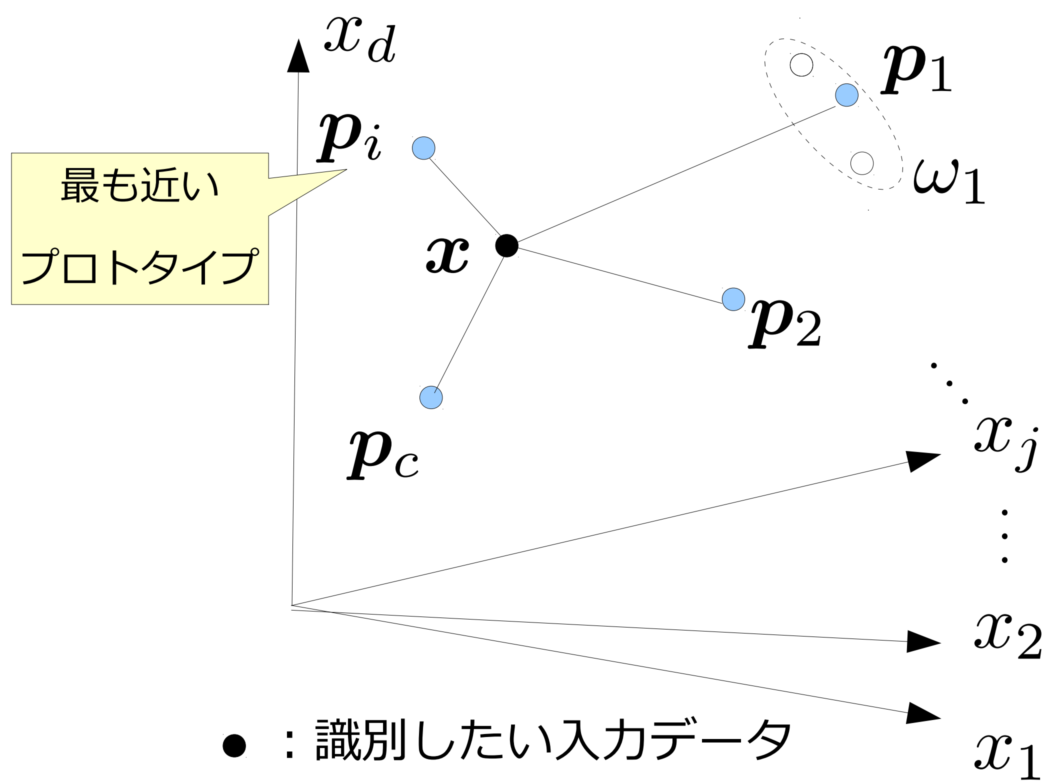
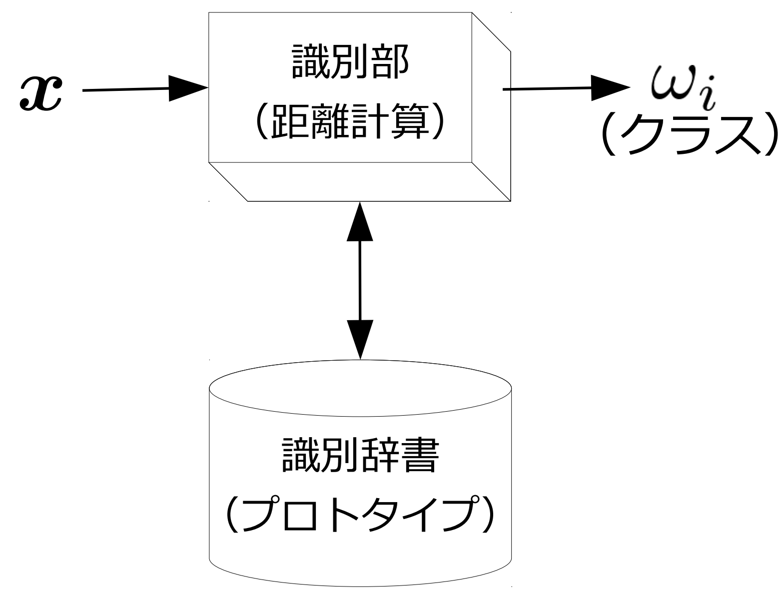
1. パターン認識システムの構成



1. 最近傍決定則（NN 法）





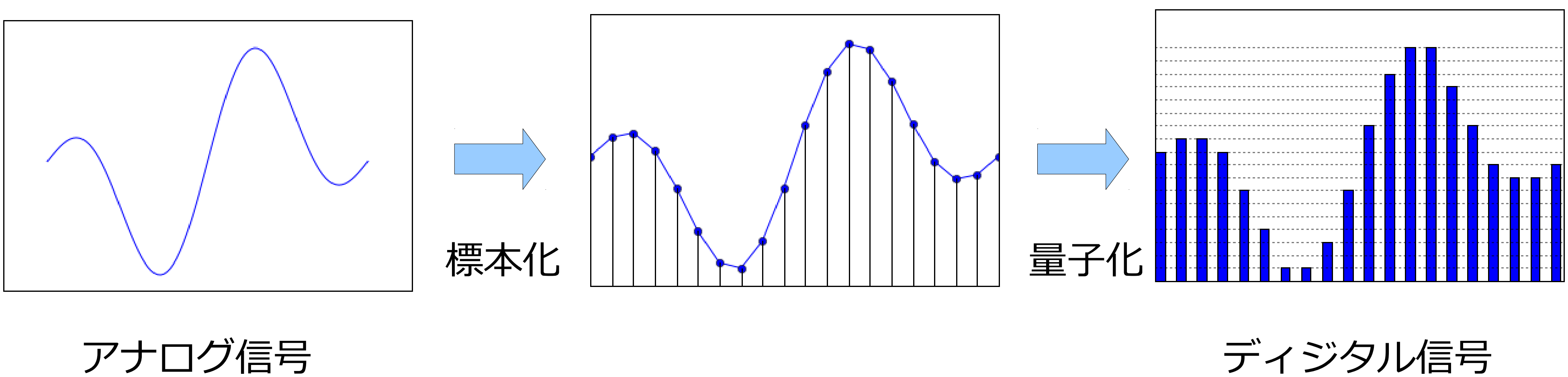
$$\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^T$$

$$\boldsymbol{p}_i = (p_{i1}, p_{i2}, \ldots, p_{id})^T$$

$$D(\boldsymbol{x}, \boldsymbol{p}_i) = ||\boldsymbol{x} - \boldsymbol{p}_i|| = \sqrt{(x_1 - p_{i1})^2 + (x_2 - p_{i2})^2 + \cdots + (x_d - p_{id})^2}$$

2. アナログ信号のディジタル化

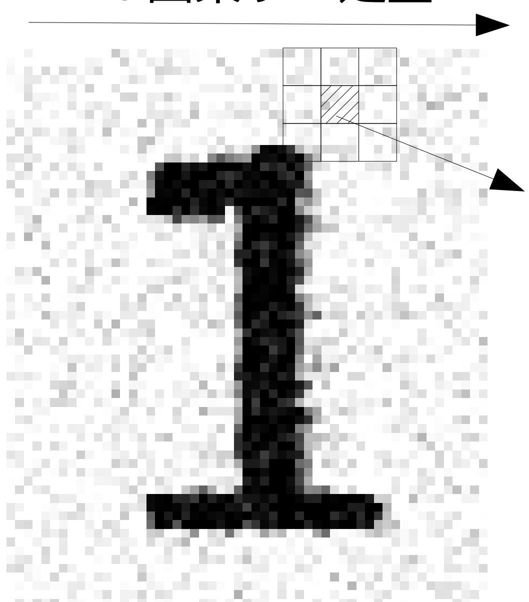
- 標本化と量子化
  - 細かすぎず、粗すぎず



2. 特徴抽出をしやすくする処理

- フィルタの適用

1 画素ずつ走査



この画素の値を

$$\sum_{p=0}^{2}\sum_{q=0}^{2} x_{i+p,j+q} h_{pq}$$

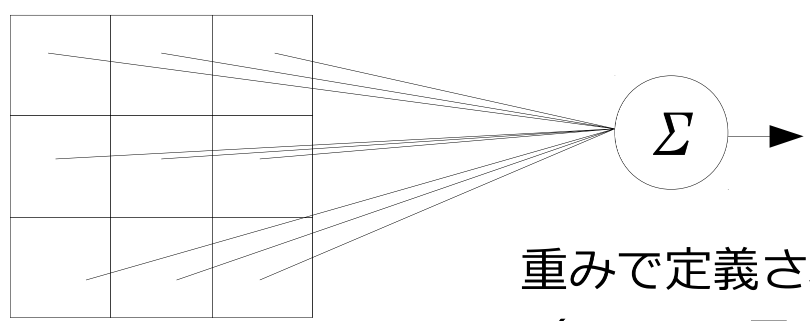
と置き換える

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

平均値フィルタ

-1	0	1
-1	0	1
-1	0	1

（縦）エッジフィルタ



重みで定義されたパターンを検出

（ CNN の畳み込み処理）

3. 特徴抽出部

- 特徴抽出部の入出力
  - 入力：ディジタル信号
  - 出力：パターンの特徴を表す d 次元ベクトル

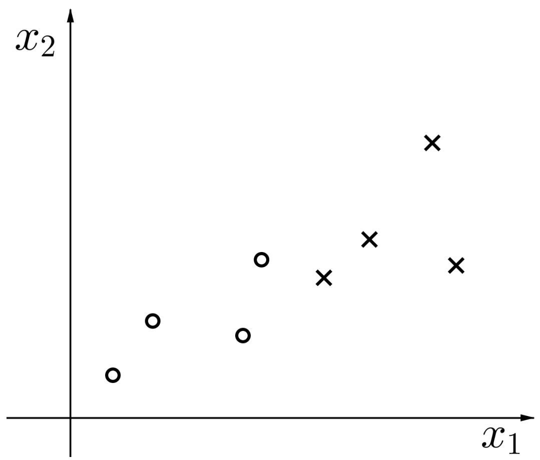
$$\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^T$$

- 特徴抽出処理
  - パターンの変動に影響されにくい特徴を選ぶ
  - 各軸のスケールを揃える：標準化処理

$$x_i' = \frac{x_i - m_i}{\sigma_i}$$

$m_i,\ \sigma_i$ ：軸 i の平均、標準偏差

3. 主成分分析

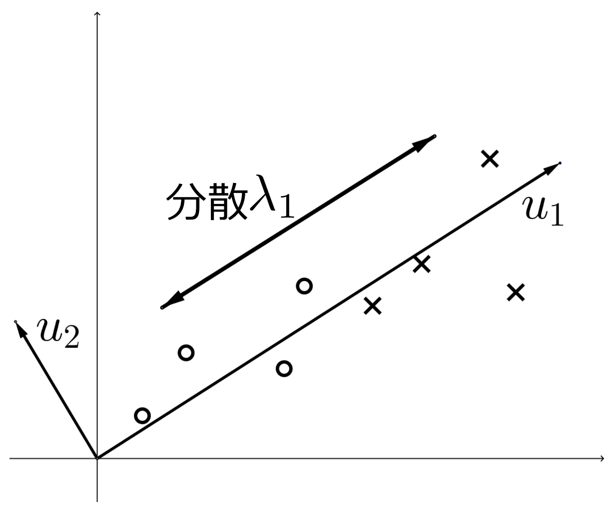


共分散行列Σの計算

$\bar{x}_1, \bar{x}_2$:平均値、 N : データ数

対角成分は分散、
非対角成分は相関を表す

$$\Sigma = \frac{1}{N}\begin{pmatrix} \sum(x_1-\bar{x}_1)^2 & \sum(x_1-\bar{x}_1)(x_2-\bar{x}_2) \\ \sum(x_1-\bar{x}_1)(x_2-\bar{x}_2) & \sum(x_2-\bar{x}_2)^2 \end{pmatrix}$$



$\boldsymbol{\Sigma}$は

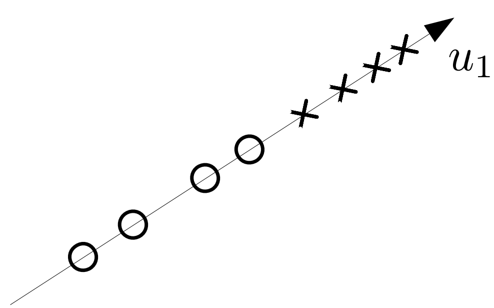
半正定値(→固有値がすべて0以上の実数)

対称行列(→固有ベクトルが実数かつ直交)

であるので、以下のように分解できる

$$\Sigma' = U^T \Sigma U = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

λは固有値の大きい順、Uは対応する
固有ベクトルU_1, U_2を並べたもの



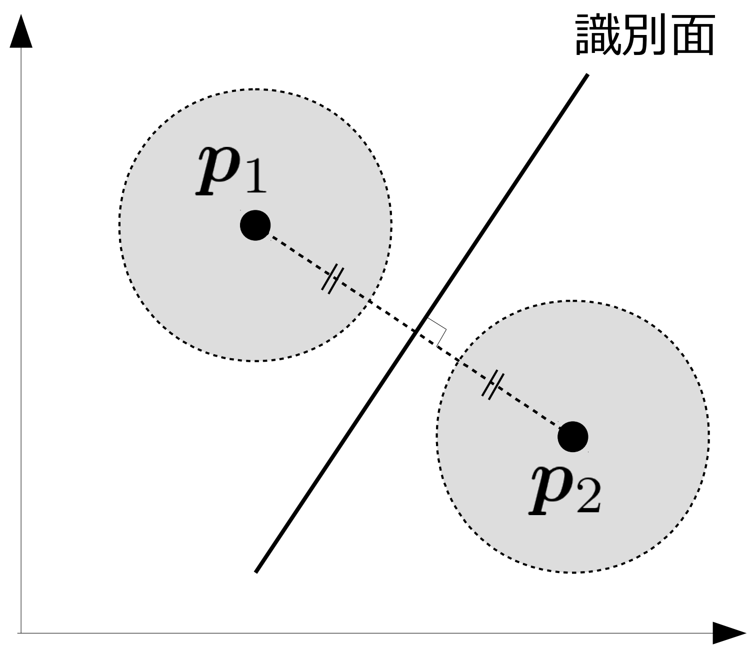
λ_1に対応する固有ベクトルU_1で
2次元データを1次元に射影

$$u_1 = U_1^T \boldsymbol{x}$$

寄与率= $\dfrac{\lambda_1}{\lambda_1 + \lambda_2}$

4. プロトタイプと識別面の関係

- 特徴空間の分割
  - クラスを分離する境界
    …プロトタイプから等距離にある領域
    - ２次元の NN 法では垂直２等分線
    - 多次元では超平面
    - 決定境界あるいは**識別面**と呼ぶ
  - 直線（超平面）で分割できる場合を**線形分離可能**と呼ぶ



4.　識別関数の設定

- １クラス１プロトタイプの NN 法の定式化
  - クラス：$\omega_1, \ldots, \omega_c$
  - プロトタイプ：$\boldsymbol{p}_1, \ldots, \boldsymbol{p}_c$
  - 入力パターン：$\boldsymbol{x}$ (特徴ベクトル)
  - NN 法：$D(\boldsymbol{x}, \boldsymbol{p}_i) = \|\boldsymbol{x} - \boldsymbol{p}_i\|$ を最小にする i を探す

$$\to \|\boldsymbol{x} - \boldsymbol{p}_i\|^2 = \|\boldsymbol{x}\|^2 - 2\boldsymbol{p}_i^T\boldsymbol{x} + \|\boldsymbol{p}_i\|^2$$

$$\to g_i(\boldsymbol{x}) = \boldsymbol{p}_i^T\boldsymbol{x} - \frac{1}{2}\|\boldsymbol{p}_i\|^2$$ を最大にする i を探す

$$\to g_i(\boldsymbol{x}) = w_{i0} + \sum_{j=1}^{d} w_{ij}x_j = \boldsymbol{w}_i^T\boldsymbol{x}$$

d+1 次元

$\boldsymbol{x} \in \omega_i$ について、$g_i(\boldsymbol{x})$ が最大になるように $\boldsymbol{w}$ を調整すればよい

4. パーセプトロンの学習アルゴリズム

- 2 クラス識別で $g(\boldsymbol{x})=g_1(\boldsymbol{x})-g_2(\boldsymbol{x})=\boldsymbol{w}^T\boldsymbol{x}$ と定義
  1. $\boldsymbol{w}$ の初期値を適当に決める
  2. 学習パターンからひとつ $\boldsymbol{x}$ を選び、 $g(\boldsymbol{x})$ を計算
  3. 誤識別が起きたときのみ、 $\boldsymbol{w}$ を修正

$$\boldsymbol{w}' = \boldsymbol{w} + \rho\boldsymbol{x}$$ （クラス 1 のパターンをクラス 2 と誤ったとき）

$$\boldsymbol{w}' = \boldsymbol{w} - \rho\boldsymbol{x}$$ （クラス 2 のパターンをクラス 1 と誤ったとき）

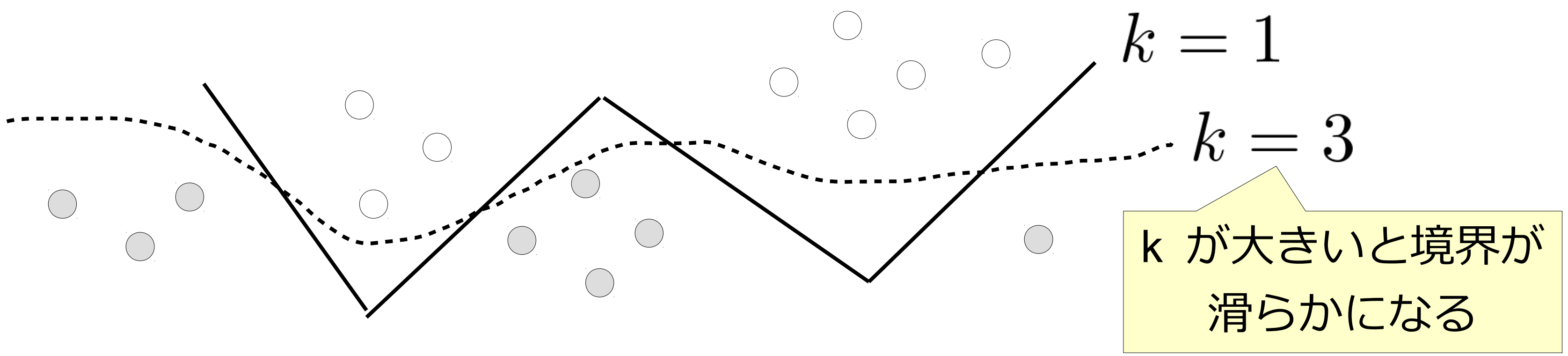
学習係数

4. 2,3 を全ての学習パターンについて繰り返す
5. すべて識別できたら終了。そうでなければ 2 へ

学習データが線形分離可能な場合は、識別面を見つけて終了

4. k-NN 法

- k- ＮＮ法とは
  - 全ての学習データをプロトタイプとする
  - 入力に近い順からｋ個のプロトタイプのクラスを調べ、多数決を取る
  - 入力への近さを重みとした重み付き多数決を用いる場合もある



5. 誤差評価に基づく学習

- 2 クラス問題を考える場合
  - 誤差関数

$$J(\boldsymbol{w}) = \frac{1}{2}\sum_{p=1}^{n}(\boldsymbol{w}^T\boldsymbol{x}_p - b_p)^2$$

最小二乗学習

- 教師信号 b_p は $\boldsymbol{x}_p \in \omega_1$ のとき 1 、 $\boldsymbol{x}_p \in \omega_2$ のとき -1

  - パターン行列 $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T$

    教師信号ベクトル $\boldsymbol{b} = (b_1, \ldots, b_n)^T$

    とすると $J(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{b}\|^2$

5. 誤差評価に基づく学習

- 解析的な解法
  - $J(\boldsymbol{w})$ が最小となる $\boldsymbol{w}$ を求める

$$\frac{\partial J}{\partial \boldsymbol{w}} = \boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{b}) = 0$$

$$\Leftrightarrow \quad \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = \boldsymbol{X}^T\boldsymbol{b}$$

$$\Leftrightarrow \quad \boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{b}$$

最小二乗法

- 最急降下法

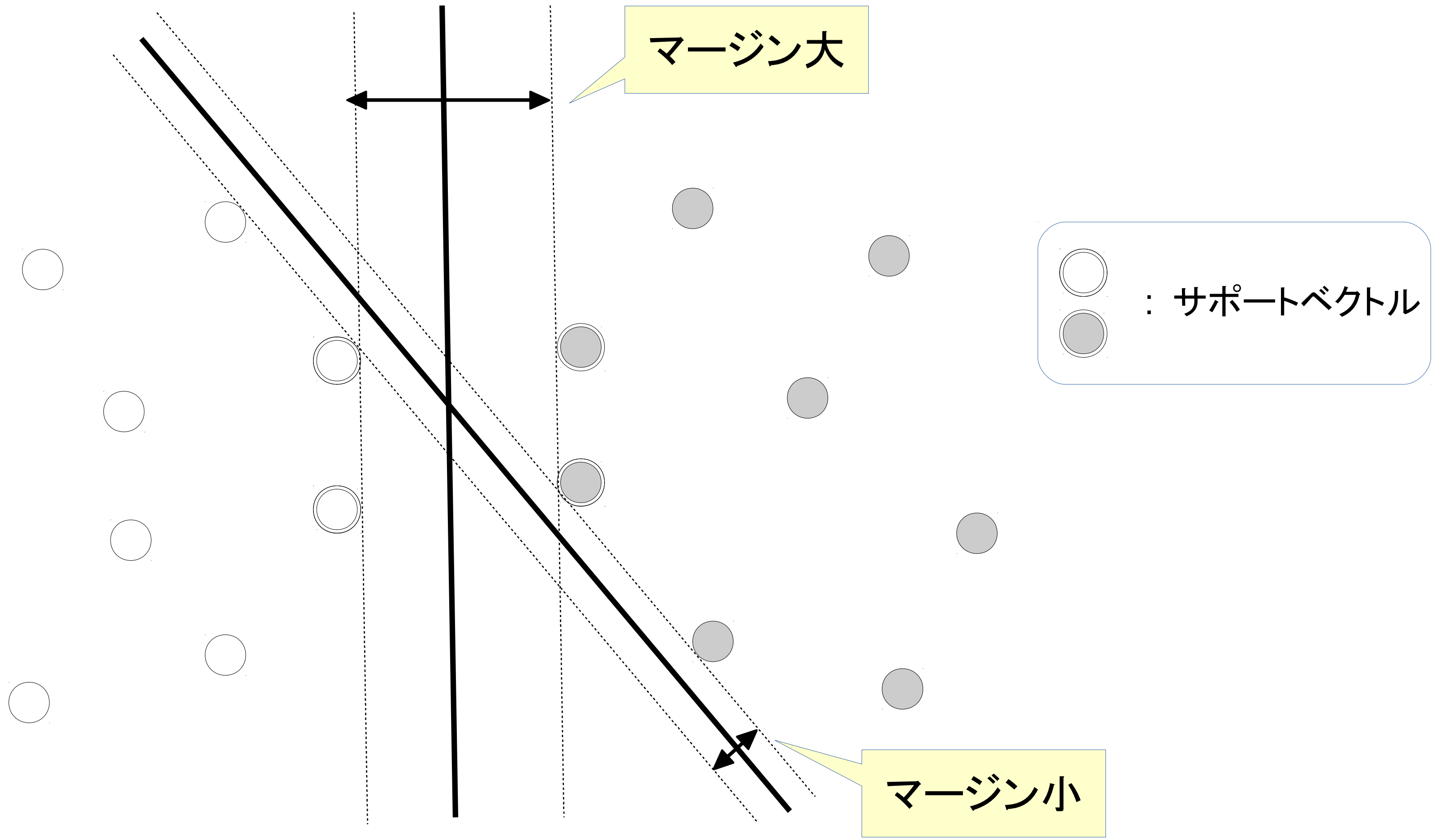
$$\boldsymbol{w}' = \boldsymbol{w} - \rho\frac{\partial J}{\partial \boldsymbol{w}}$$

$$= \boldsymbol{w} - \rho\sum_{p=1}^{n}(\boldsymbol{w}^T\boldsymbol{x}_p - b_p)\boldsymbol{x}_p$$

<u>確率的最急降下法</u>
　各データに対して更新
<u>ミニバッチ法</u>
　適切な個数でまとめて更新

6. SVM

- 線形 SVM
  - マージン最大となる線形識別面を求める



6. SVM

- 学習データ

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

- 線形識別面の式

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

- 識別面の制約の導入（係数を定数倍しても平面は不変）

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

- 学習パターンと識別面との最小距離（＝マージン）

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

これを
最大化

点と直線の距離の公式

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

6. SVM

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

極値が求め
やすい２乗に

- 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 $L(x, \alpha) = f(x) - \alpha g(x)$
    - $\alpha \geq 0$
    - x, α で偏微分して０になる値が極値

6. SVM

- 計算

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

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

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

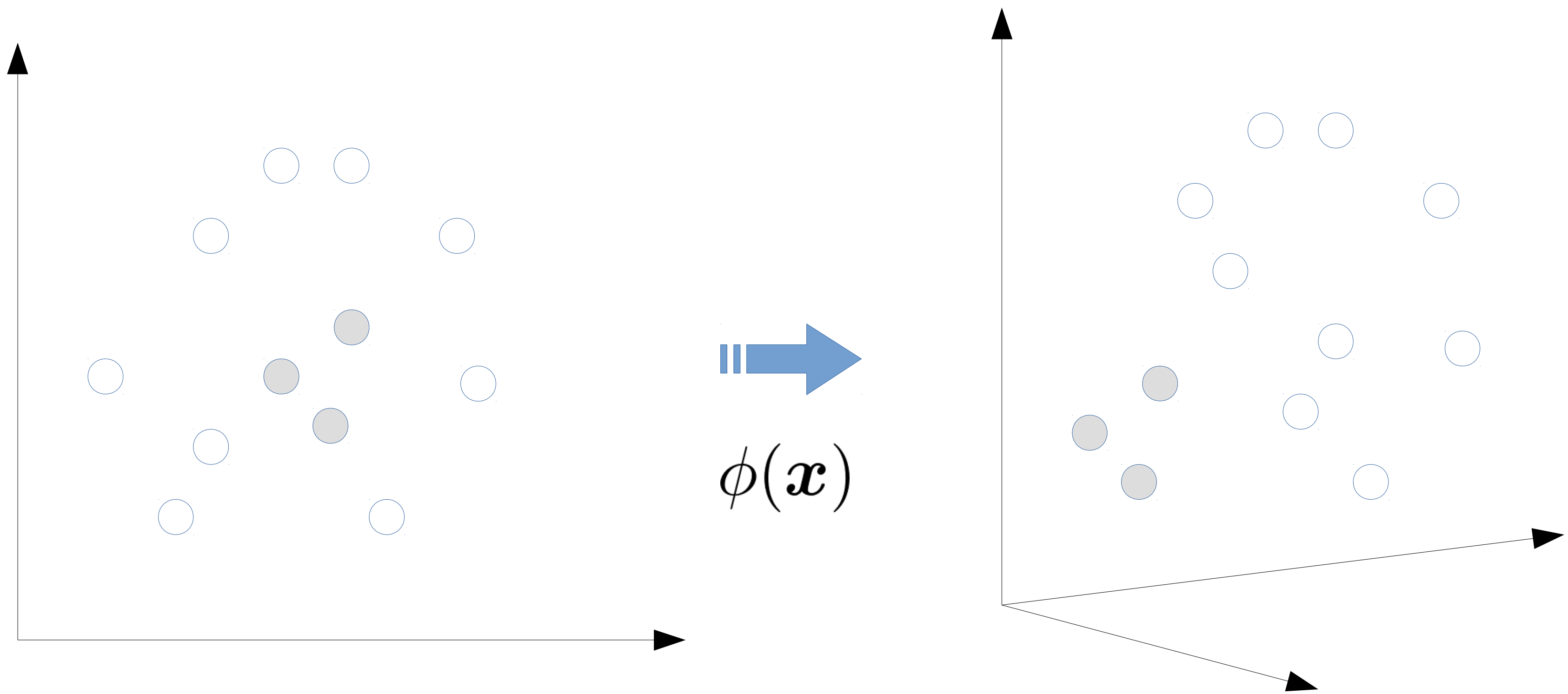
$$L(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j - \sum_{i=1}^{n} \alpha_i$$

$$\alpha_i \geq 0$$

最大化 2 次計画問題
$\rightarrow \boldsymbol{\alpha}$ が求まる

6. SVM

- 特徴ベクトルの次元数を増やす



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

6. SVM

- 非線形変換関数： $\phi(\boldsymbol{x})$
- カーネル関数
  - 元の空間での距離が変換後の空間の内積に対応

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

線形カーネル
$(\boldsymbol{x}^T \boldsymbol{x}')^p$
を用いる場合もある

  - カーネル関数の例
    - 多項式カーネル $K(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^T \boldsymbol{x}' + 1)^p$
    - ガウシアンカーネル $K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\dfrac{||\boldsymbol{x} - \boldsymbol{x}'||^2}{\sigma^2})$

この形であれば、対応する非線形変換が
存在することが数学的に保証されている

6. 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

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

非線形変換の
式は不要！！！

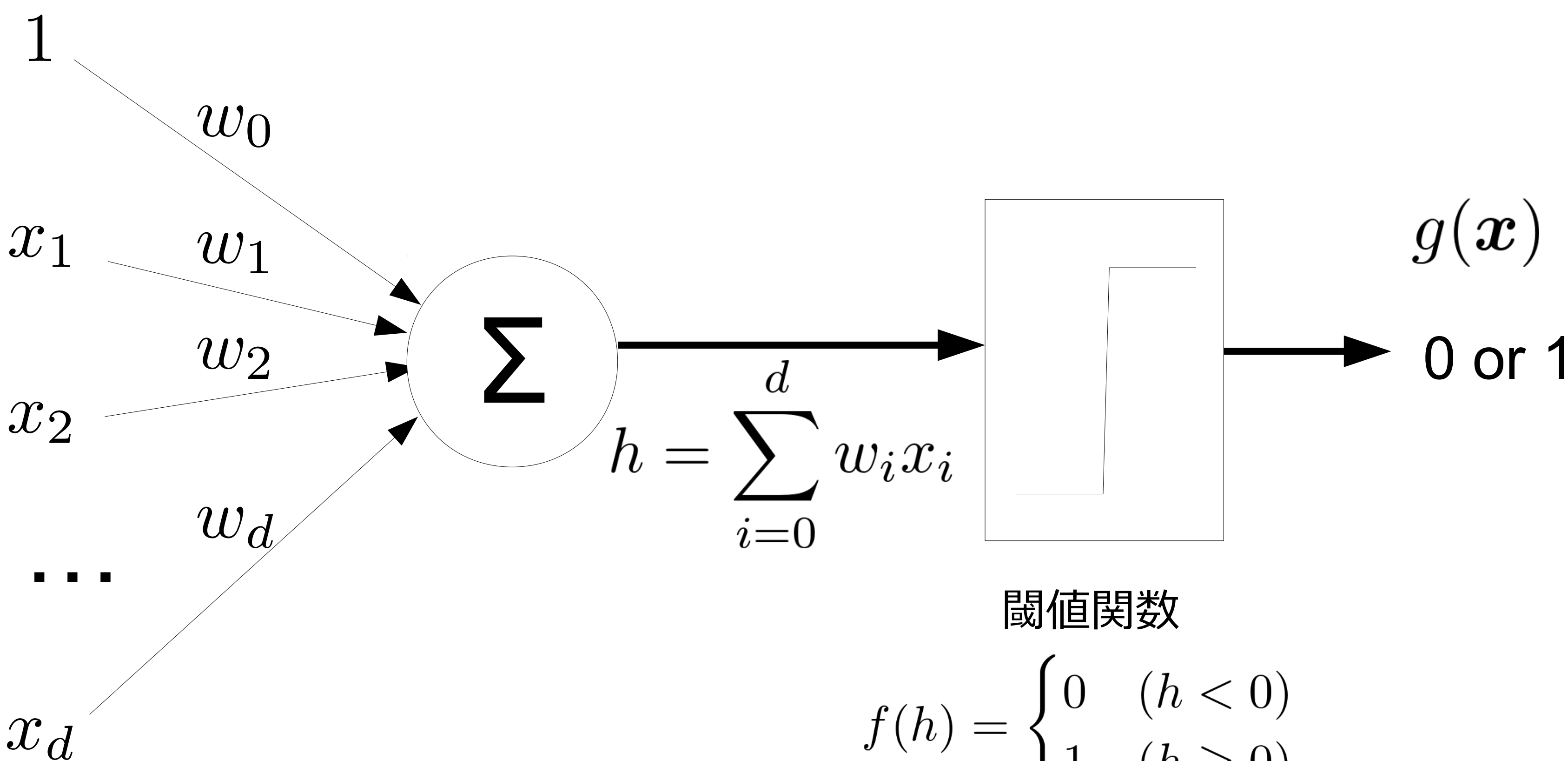
カーネルトリック

7. ニューラルネットワーク

- 単層パーセプトロンの定義

以後、 $\boldsymbol{w}$ は w_0 を含む

  - $\boldsymbol{w}^T \boldsymbol{x} = 0$ という特徴空間上の識別面を表現



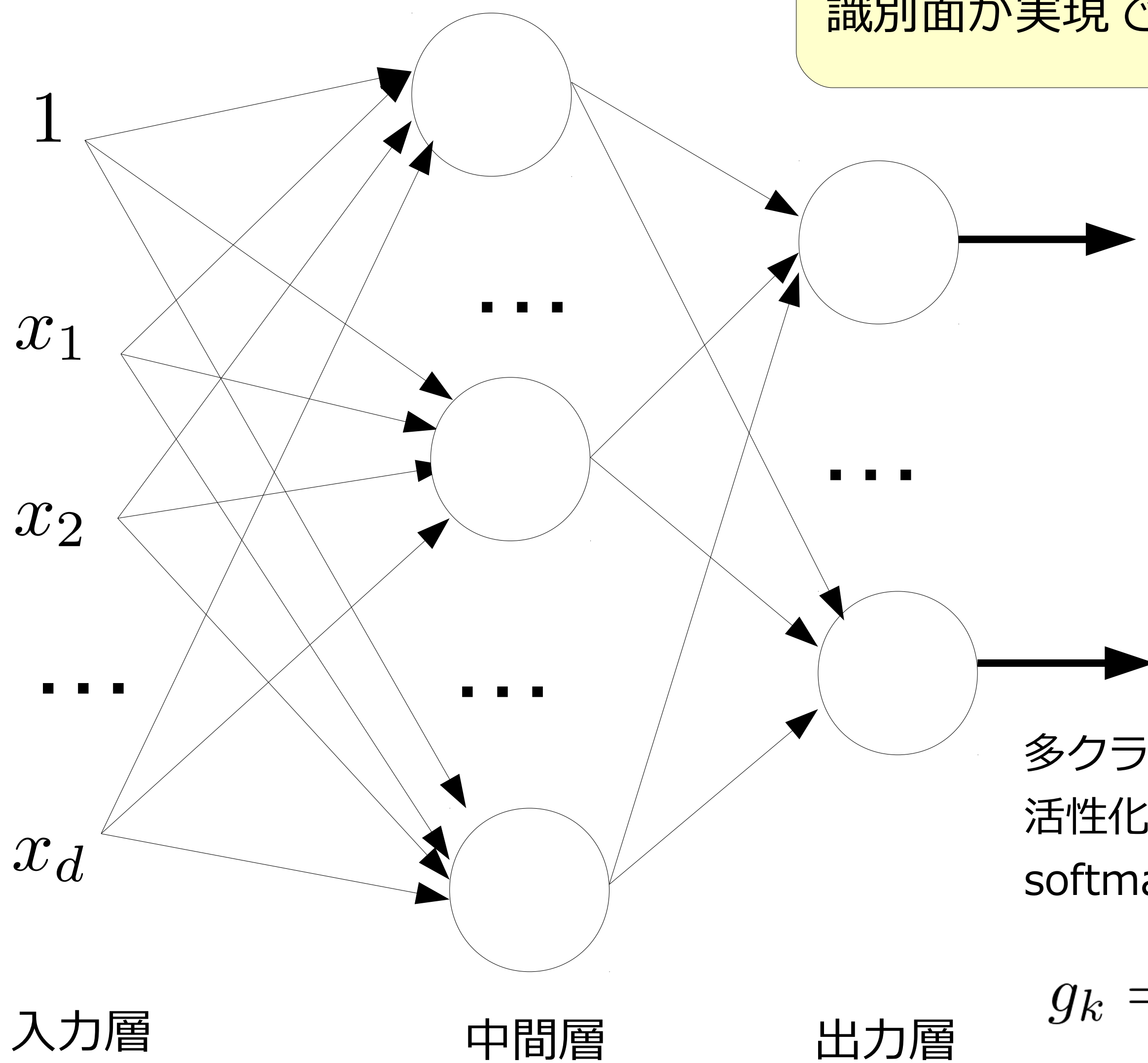
閾値関数

$$f(h) = \begin{cases} 0 & (h < 0) \\ 1 & (h \geq 0) \end{cases}$$

7. ニューラルネットワーク

- 多層パーセプトロン

特徴空間上で複雑な非線形識別面が実現できる

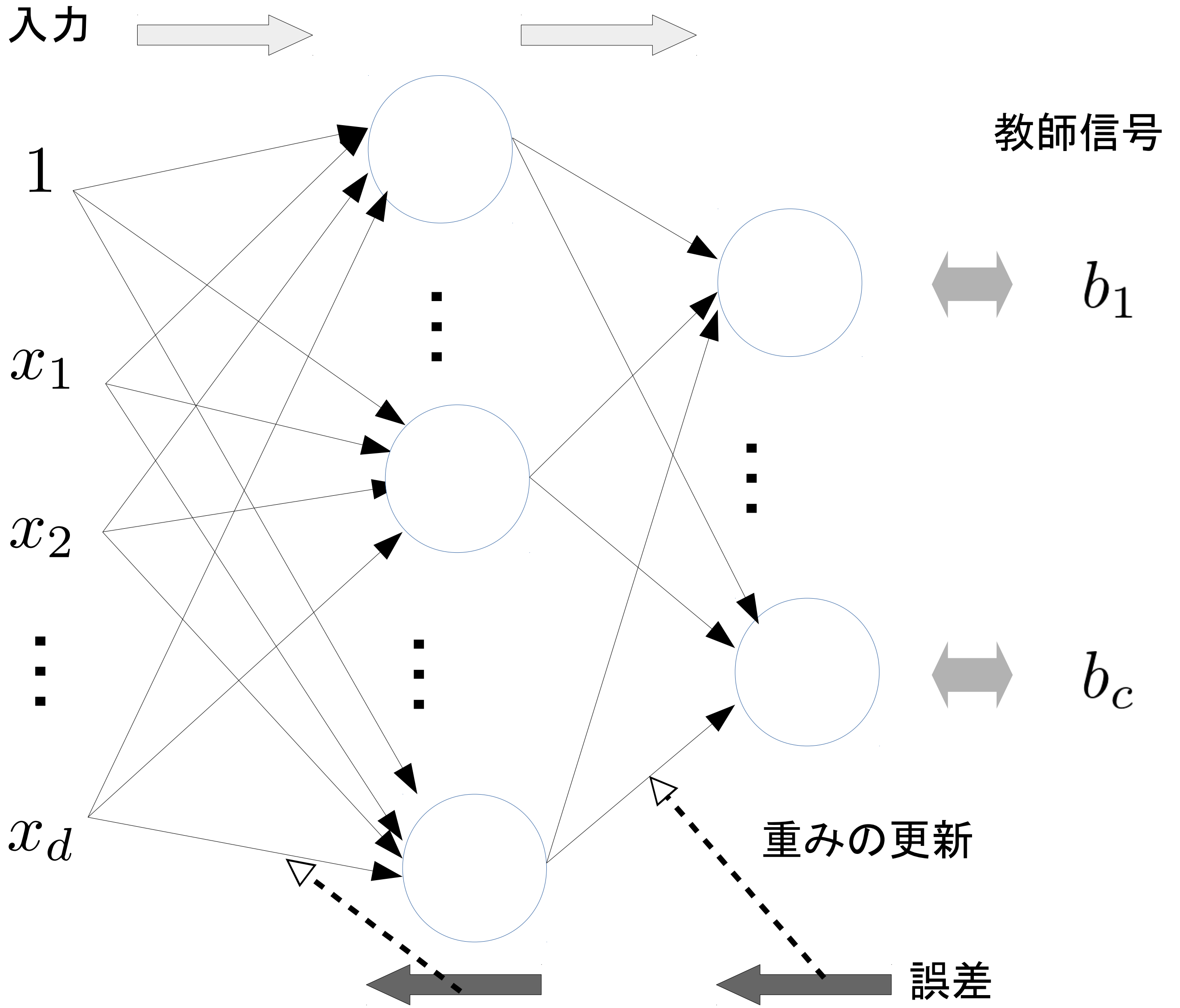


多クラス問題の出力層には活性化関数として以下のsoftmax 関数を用いる

$$g_k = \frac{\exp(h_k)}{\sum_{j=1}^{c} \exp(h_j)}$$

7. ニューラルネットワーク

- 誤差逆伝播法



7. ニューラルネットワーク

1. リンクの重みを小さな初期値に設定
2. 個々の学習データ $(\boldsymbol{x}_p, b_p)$に対して以下繰り返し

   a)入力 $\boldsymbol{x}_p$ に対するネットワークの出力 $\boldsymbol{g}_p$ を計算

   b)出力層の k 番目のユニットに対してエラー量 ε を計算

   $$\varepsilon_k \leftarrow (g_k - b_k) g_k (1 - g_k)$$

   c)中間層の h 番目のユニットに対してエラー量 ε を計算

   $$\varepsilon_j \leftarrow \left(\sum_k \varepsilon_k w_k\right) g_j (1 - g_j)$$

   d)重みの更新

   $$w_{ji} \leftarrow w_{ji} + \rho \varepsilon_j x_{pj}$$

局所最適解の可能性が
高いので、初期値を変
えて繰り返す

8. 統計的手法

- 事後確率最大法（ベイズ決定則）
  - $P(\omega_i \mid \boldsymbol{x})$ を最大にするクラス ω_i を識別結果とする

$$
\begin{aligned}
&\arg\max_{i=1,\dots,c} P(\omega_i|\boldsymbol{x}) \\
&= \arg\max_{i=1,\dots,c} \frac{p(\boldsymbol{x}|\omega_i)P(\omega_i)}{p(\boldsymbol{x})} \\
&= \arg\max_{i=1,\dots,c} p(\boldsymbol{x}|\omega_i)P(\omega_i)
\end{aligned}
$$

ベイズの定理

8．統計的手法

- 事前確率 $P(\omega_i)$ の求め方
  - 最尤推定
    - 学習データ数：N
    - クラス ω_i のデータ数： n_i
    - 事前確率の最尤推定値

$$P(\omega_i) = \frac{n_i}{N}$$

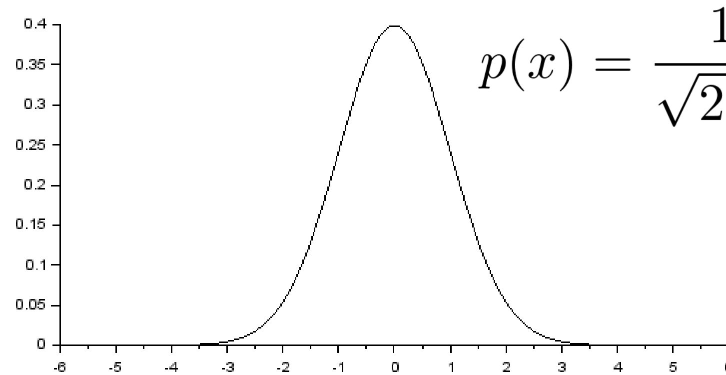
8．統計的手法

- クラス分布 $p(\boldsymbol{x}|\omega_i)$ の求め方
  - 確率分布の形を仮定して、そのパラメータを学習データから推定
  - 例）正規分布：平均と共分散行列を推定

$$p(\boldsymbol{x}|\omega_i) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^t \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)\}$$

$\boldsymbol{m}_i$ ：平均ベクトル

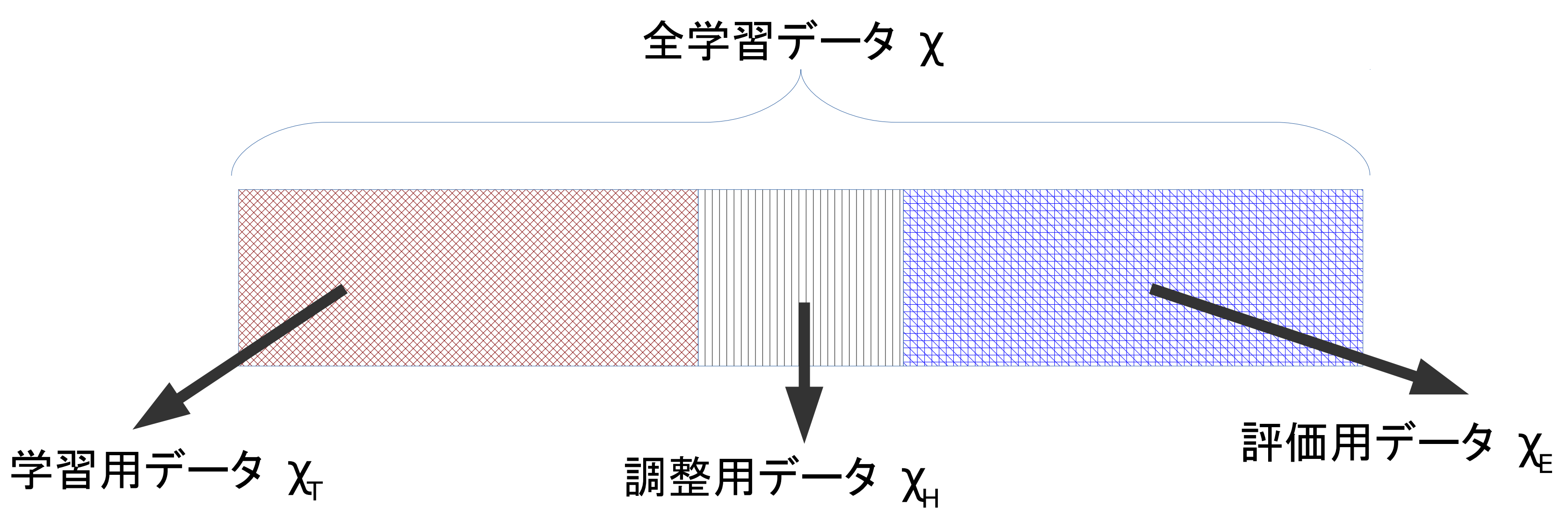
$\boldsymbol{\Sigma}_i$ ：共分散行列



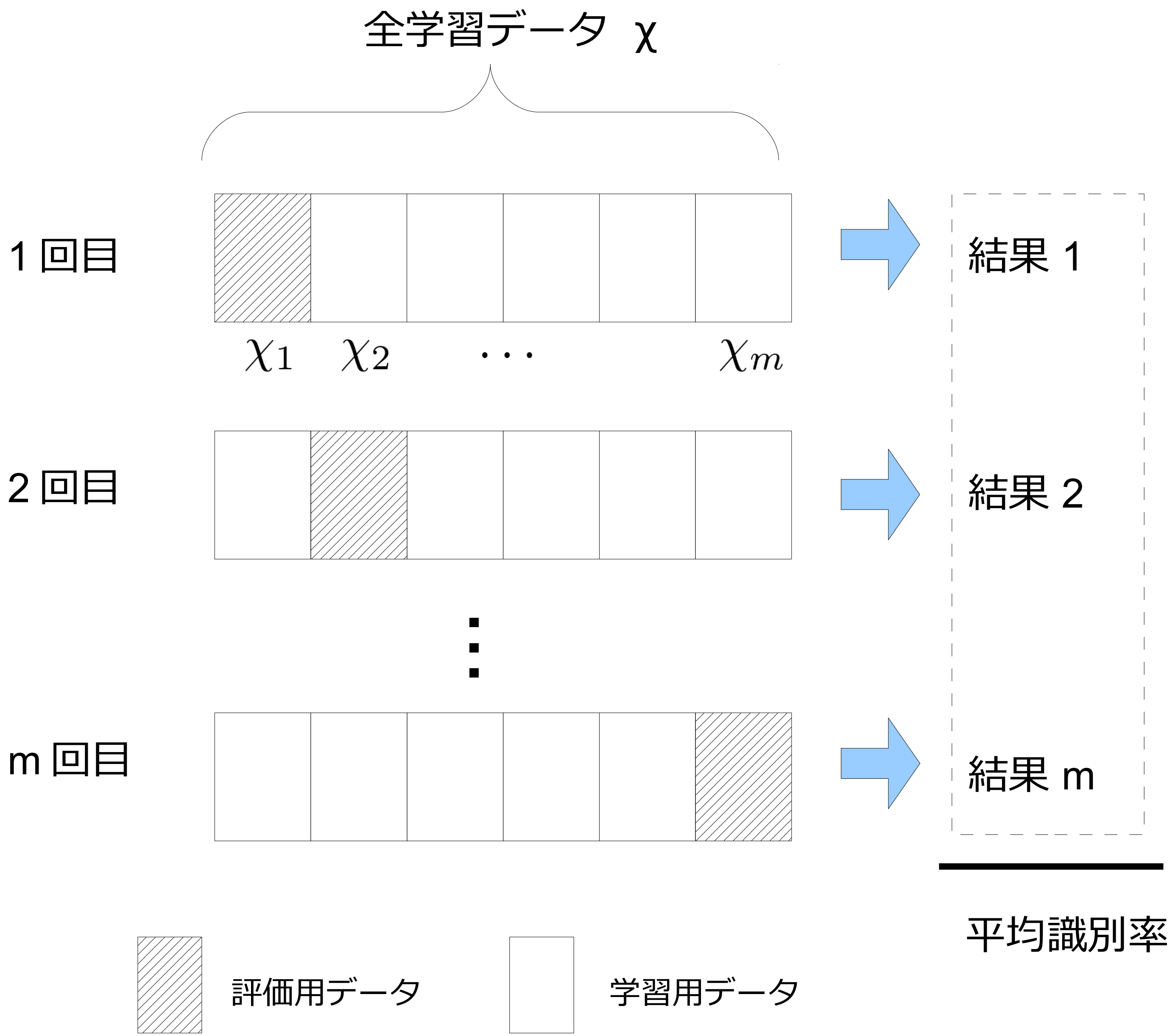
$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

9．分割学習法

- ハイパーパラメータ調整を含む分割学習法
  - 全学習データ χ を学習用データ集合 χ_T、調整用データ集合 χ_H、評価用データ集合 χ_E に分割する
  - χ_T を用いて識別機を設計、 χ_H を用いてハイパーパラメータを調整、 χ_E を用いて誤識別率を推定する



9．交差確認法



全学習データ χ
1 回目
χ_1 χ_2 $\cdots$ χ_m
2 回目
m 回目
結果 1
結果 2
結果 m
平均識別率
評価用データ
学習用データ

9. ハイパーパラメータ調整

- パラメータ ➡ 学習可能
  - 識別関数の重み
  - ニューラルネットワークの結合の重み
  - k-NN 法のプロトタイプの位置
- ハイパーパラメータ ➡ 学習結果によって調整
  - 識別関数の次数
  - ニューラルネットワークの中間ユニット数
  - k-NN 法の k

9. ハイパーパラメータ調整

- 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 e_λ を求める