認識行動システムの基礎

橋本浩一

koichi@k2.t.u-tokyo.ac.jp

情報理工学系研究科システム情報学専攻

○ パターン認識とは
○ 線形代数の復習・座標変換
○ ロボットキネマティクス・ステレオ
○ 多変量解析
○ パーセプトロン
○ 画像パターンの認識

シラバス

- 認識行動システムを構築するにあたり基礎的な事項を整理して講義する．
- まず，行動系の設計・解析に必要となる座標変換について述べ，ロボットキネマティクスの基礎を学習する．次に射影透視変換を導入し，座標変換，射影透視変換などが同次変換と呼ばれる線形変換により統一的に表現できることを示す．さらに，ステレオ視による3次元推定問題に触れる．
- 一方，認識システムの基礎として，多変量解析の基礎を説明し，特徴空間を導入する．そして最後に，パーセプトロンを代表とするパターン認識システムと学習アルゴリズムを紹介する．
- 本講義を通して，線形代数の成果がさまざまな形態で応用されることを学ぶであろう．線形代数をきちんと復習しておくこと．
- 成績は主に試験により判断するが，レポートおよび出席も考慮する．

参考書

1. G. ストラング著，線形代数とその応用，産業図書
2. 吉川著，ロボット制御基礎論，コロナ社
3. 出口著，ロボットビジョンの基礎，コロナ社
4. 浅野著，入門多変量解析の実際第2版，講談社
5. 石井，上田，前田，村瀬共著，パターン認識，オーム社

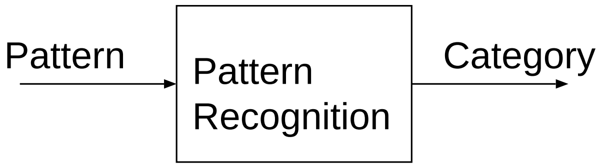
パターン認識とは

- パターン認識: 「パターンをそれが属すべきカテゴリに対応づける操作」
  - **パターン**: 「空間的または時間的に観測可能な事象であって，観測された対象どうしが同一であるか (または似ているか) 否かを判定できるような性質を備えているもの」
  - **カテゴリ** (または**クラス**): 「パターン認識の結果，同等とみなされるパターンの集合概念」
- 「対象に認められる特徴要素の単なる集まりでなく，各特徴要素の間に認められる位置的，時間的，機能的関連までを含んで，これらの共通的性質全体を**パターン**と呼び，対象にそのパターンの存在を認めることを**パターン認識**という」
- 例: 音声認識，文字認識，物体認識，画像認識...
- 研究の動機: 人間が行っているような認識機能を機械でも実現したい
- 目的: **機械による認識の実現**

パターン認識の研究

- パターン認識の機能 (メカニズム) の解明
  - 現存する最良の見本である人間 (またはより広く生体) のパターン認識機能の研究
  - 生物学，生理学，心理学
  - 生体機能のモデル
  - バイオニクス，バイオサイバネティクス，ニューラルネット
- 機械によるパターン認識機能の実現方法の開発
  - 生体の機能からヒントを得ることもある
  - 数理的手法
  - コンピュータによる実現，アルゴリズム
  - 多変量解析，ニューラルネット，パターンマッチング

パターン認識の過程



- 入力パターン (現象，信号)
- 出力カテゴリ (概念，記述)
- 中間過程
  - **観測**: 入力パターンを認識系の内部に取り込む操作
  - **前処理**: 後段の処理をしやすくするための予備的処理
  - **特徴抽出**: カテゴリの選択に役立つようなパターン特有の性質を担うごく少数の特徴量 (数値，記号) の組 (特徴パラメータ) を取り出す.
  - **決定**: 抽出された特徴パラメータの値を用いて，入力パターンが所属するとみなされるカテゴリをきめる.

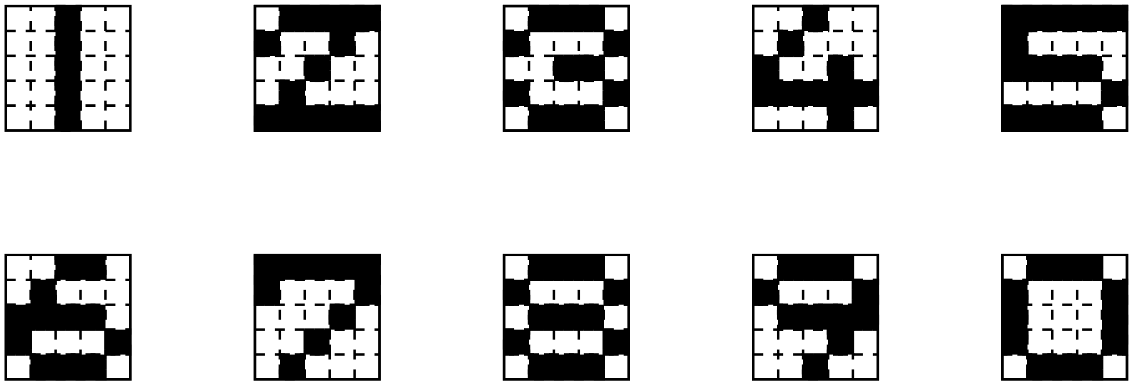
パターンの表現

- パターン空間
  - **パターンベクトル**: パターンを表す n 個の数値の組

$$\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$$

  - **パターン空間**: パターンベクトルのつくる空間．もともとよく似ていたパターンはパターン空間の中でも近くにある．
- 特徴空間
  - **特徴ベクトル**: パターンに関する本質的な情報を含む少数個の数値の組．**位相**を崩さずに次元数を極力減らす．

特徴ベクトルと特徴空間の例



5 × 5 メッシュによる 2 値パターン

- パターン

$$\boldsymbol{x} = (x_1, x_2, \cdots, x_{25})$$

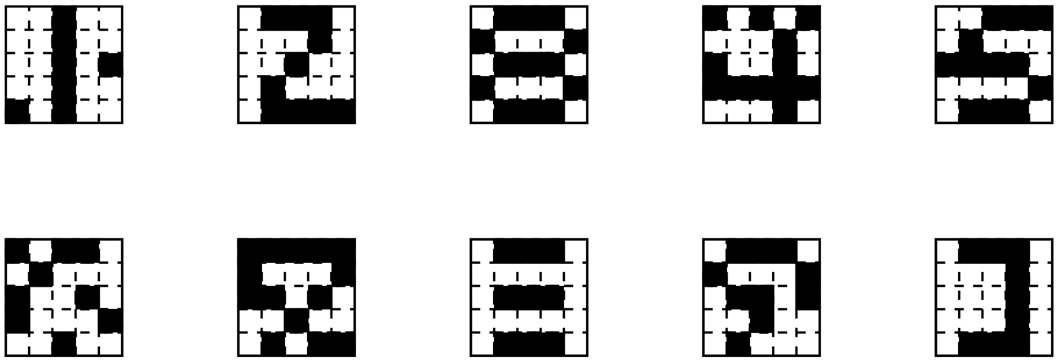
- 特徴量

$$\begin{cases} x_j = 1 & (\text{黒: 文字部分}) \\ x_j = 0 & (\text{白: 背景部分}) \end{cases} \quad (1 \leq j \leq 25)$$

- パターンの組み合わせ: $2^{25} = 33554432$ 通り
- 出力カテゴリ: 0 から 9 までの数字

$$\Omega = \{\omega_1, \omega_2, \cdots, \omega_{10}\} \quad (c = 10)$$

識別アルゴリズム



入力パターン

- **全数記憶方式**: 33554432 通りのすべてのパターンの参照テーブルをつくる.
  - もっとも簡単な識別法
  - 記憶容量，テーブル作成作業が膨大

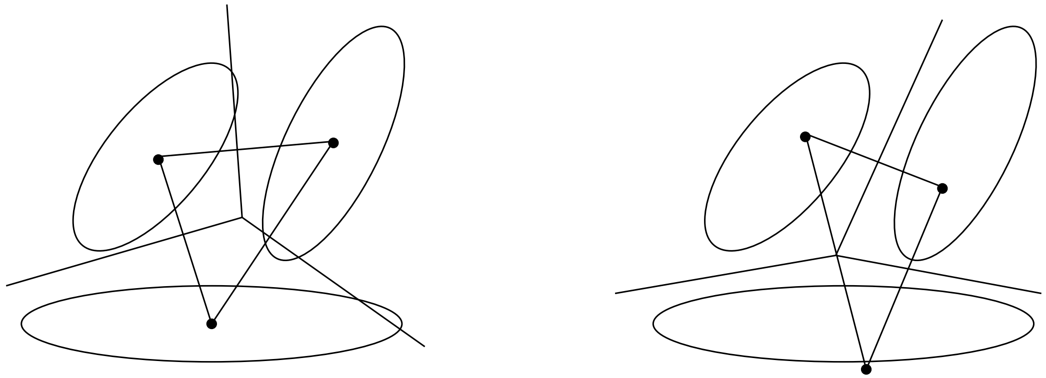
もう少し効率の良さそうなアルゴリズム

- **最近傍決定則**: 入力パターンともっとも距離の近いプロトタイプの属するカテゴリを選ぶ.

$$\min_{p=1,\cdots,n}\{D(\boldsymbol{x},\boldsymbol{x}_p)\} = D(\boldsymbol{x},\boldsymbol{x}_k) \;\Rightarrow\; \boldsymbol{x} \in \omega_k$$

  – 特徴空間の分割法 (3 と 8 は似ている)

  – プロトタイプの選び方 (6 と回転した 9 の識別)



領域分割とプロトタイプ

線形識別関数

- 1 カテゴリあたり 1 プロトタイプ
- c 個のカテゴリ $\omega_1, \cdots, \omega_c$ のプロトタイプとして，d 次元ベクトル $\boldsymbol{p}_1, \cdots, \boldsymbol{p}_c$ を選んだとする．
- プロトタイプと入力パターンの距離

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

- 最近傍決定則

$$\max_{i=1,\cdots,c}\{g_i(\boldsymbol{x})\} = g_k(\boldsymbol{x}) \;\Rightarrow\; \boldsymbol{x} \in \omega_k$$

$$g_i(\boldsymbol{x}) = \boldsymbol{p}_i^T\boldsymbol{x} - \frac{1}{2}\|\boldsymbol{p}_i\|^2$$

- 線形識別関数の例

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

$$\boldsymbol{w}_i = [w_{i0}, w_{i1}, \cdots, w_{id}]^T, \quad \boldsymbol{x} = [1, \boldsymbol{x}^T]^T$$

- 1 カテゴリあたり 1 プロトタイプの最近傍決定則は線形識別関数による識別法と等価

線形識別器

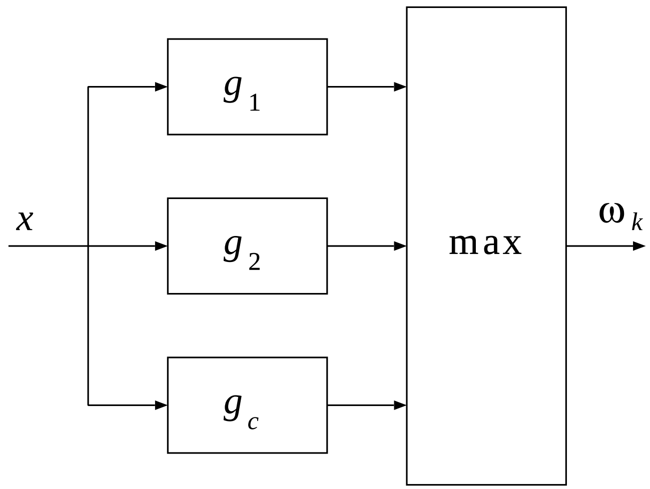
- 学習パターン全体の集合: $\mathcal{X}_i (i = 1, \cdots, c)$
- **学習**: $\mathcal{X}_i$ に属するすべての $\boldsymbol{x}$ に対して

$$g_i(\boldsymbol{x}) > g_j(\boldsymbol{x}) \qquad (g_i(\boldsymbol{x}) = \boldsymbol{w}_i\boldsymbol{x}, \quad j = 1, \cdots, c,\ j \neq i)$$

  が成り立つように重み $\boldsymbol{w}$ を決定すること.

- 識別アルゴリズム:

$$\max_{i=1,\cdots,c} \{g_i(\boldsymbol{x})\} = g_k(\boldsymbol{x}) \ \Rightarrow\ \boldsymbol{x} \in \omega_k$$



識別器

パーセプトロン

- 重みベクトル $\boldsymbol{w}_i$ の初期値を適当に設定する.
- $\mathcal{X}$ のなかから学習パターンをひとつ選び，それを $\boldsymbol{x}$ とする.
- 識別関数

$$g_i(\boldsymbol{x}) = \boldsymbol{w}_i^T \boldsymbol{x}$$

  によって識別を行う．$(k = \arg\max_{i=1,..,c}\{g_i(\boldsymbol{x})\})$

- 誤識別を生じた場合 (ω_i に属するパターンを ω_j と誤ったとする) のみ次式により重みベクトルを修正する.

$$\begin{aligned}\boldsymbol{w}_i' &= \boldsymbol{w}_i + \rho\boldsymbol{x}\\ \boldsymbol{w}_j' &= \boldsymbol{w}_j - \rho\boldsymbol{x}\end{aligned}$$

- 上の処理，2, 3 を $\mathcal{X}$ の全パターンに対して繰り返す.
- $\mathcal{X}$ の全パターンを正しく識別できたら終了，誤りがあるときには 2 に戻る.

1次元の例

- カテゴリ: $\omega_1 = \{x : -2/5 < x\},\ \omega_2 = \{x : x < -2/5\}$
- 学習パターン:

pattern	value	categoly
x_1	-0.2	ω_1
x_2	-1.5	ω_2
x_3	-1.0	ω_2
x_4	0.2	ω_1
x_5	1.2	ω_1
x_6	-0.5	ω_2

1次元の例(アルゴリズム)

- 例: $\boldsymbol{w} = (2, -7), \rho = 1$

$$
\begin{array}{rclll}
g(x_1) & = & 2 - 7(-0.2) = 3.4, & & \boldsymbol{w} = (2, -7) \\
g(x_2) & = & 2 - 7(-1.5) = 12.5, & & \boldsymbol{w} = (2, -7) - (1, -1.5) = (1, -5.5) \\
g(x_3) & = & 1 - 5.5(-1.0) = 6.5, & & \boldsymbol{w} = (1, -5.5) - (1, -1.0) = (0, -4.5) \\
g(x_4) & = & 0 - 4.5(0.2) = -0.9, & & \boldsymbol{w} = (0, -4.5) + (1, 0.2) = (1, -4.3) \\
g(x_5) & = & 1 - 4.3(1.2) = -4.16, & & \boldsymbol{w} = (1, -4.3) + (1, 1.2) = (2, -3.1) \\
g(x_6) & = & 2 - 3.1(-0.5) = 3.55, & & \boldsymbol{w} = (2, -3.1) - (1, -0.5) = (1, -2.6) \\
\vdots & & & & \vdots \\
\vdots & & & & \vdots \\
g(x_1) & = & 0 + 2.3(-0.2) = -0.46, & & \boldsymbol{w} = (0, 2.3) + (1, -0.2) = (1, 2.1) \\
g(x_2) & = & 1 + 2.1(-1.5) = -2.15, & & \boldsymbol{w} = (1, 2.1) \\
g(x_3) & = & 1 + 2.1(-1.0) = -1.1, & & \boldsymbol{w} = (1, 2.1) \\
g(x_4) & = & 1 + 2.1(0.2) = 1.42, & & \boldsymbol{w} = (1, 2.1) \\
g(x_5) & = & 1 + 2.1(1.2) = 3.52, & & \boldsymbol{w} = (1, 2.1) \\
g(x_6) & = & 1 + 2.1(-0.5) = -0.05, & & \boldsymbol{w} = (1, 2.1) \\
g(x_1) & = & 1 + 2.1(-0.2) = 0.58, & & \boldsymbol{w} = (1, 2.1)
\end{array}
$$

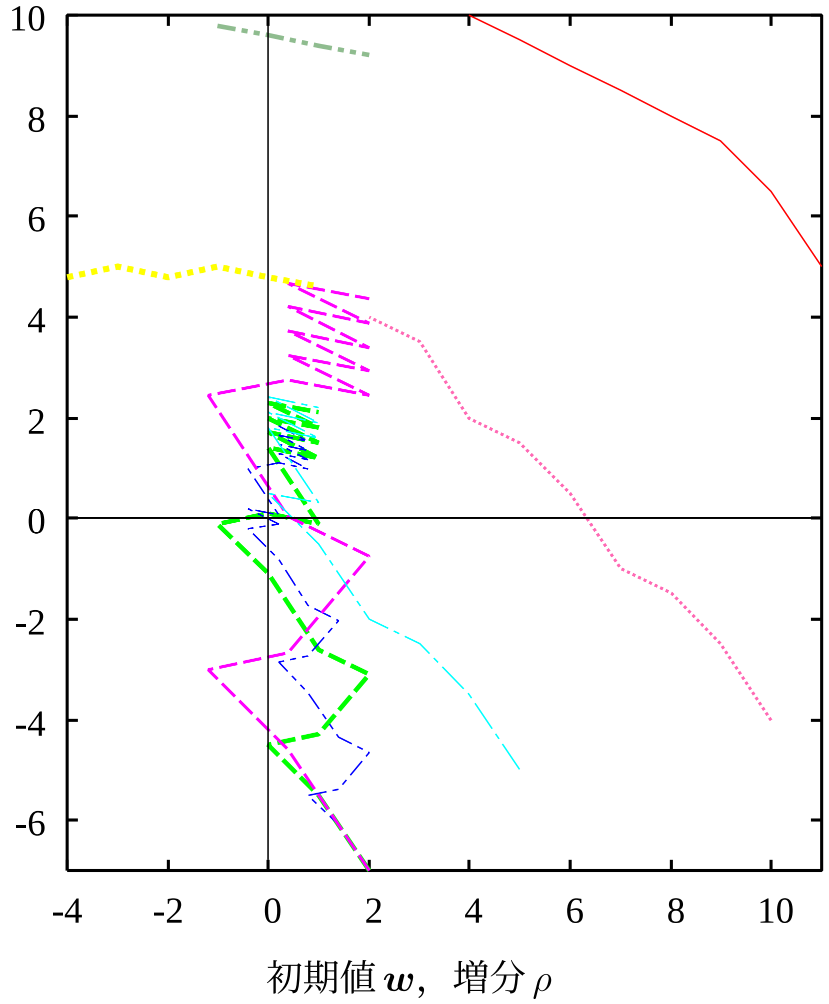
1次元の例 (2つの識別関数)

pattern	value	categoly
x_1	-0.2	ω_1
x_2	-1.5	ω_2
x_3	-1.0	ω_2
x_4	0.2	ω_1
x_5	1.2	ω_1
x_6	-0.5	ω_2

```
N i o:  g1(x)  g2(x)  w1(1)  w1(2)  w2(1)  w2(2)
------------------------------------------------
1 1 1:  3.400  0.000  2.000 -7.000  0.000  0.000
1 2 2: 12.500  0.000  2.000 -7.000  0.000  0.000
1 3 2:  6.500  2.500  1.000 -5.500  1.000 -1.500
1 4 1: -0.900  1.500  0.000 -4.500  2.000 -2.500
1 5 1: -4.160 -2.240  1.000 -4.300  1.000 -2.700
1 6 2:  3.550  1.950  2.000 -3.100  0.000 -3.900
2 1 1:  1.520  1.880  1.000 -2.600  1.000 -4.400
2 2 2:  6.200  6.300  2.000 -2.800  0.000 -4.200
2 3 2:  4.800  4.200  2.000 -2.800  0.000 -4.200
2 4 1:  0.640 -0.040  1.000 -1.800  1.000 -5.200
2 5 1: -1.160 -5.240  1.000 -1.800  1.000 -5.200
2 6 2:  1.900  3.600  1.000 -1.800  1.000 -5.200
3 1 1:  1.360  2.040  1.000 -1.800  1.000 -5.200
3 2 2:  5.000  7.500  2.000 -2.000  0.000 -5.000
3 3 2:  4.000  5.000  2.000 -2.000  0.000 -5.000
3 4 1:  1.600 -1.000  2.000 -2.000  0.000 -5.000
3 5 1: -0.400 -6.000  2.000 -2.000  0.000 -5.000
3 6 2:  3.000  2.500  2.000 -2.000  0.000 -5.000
```

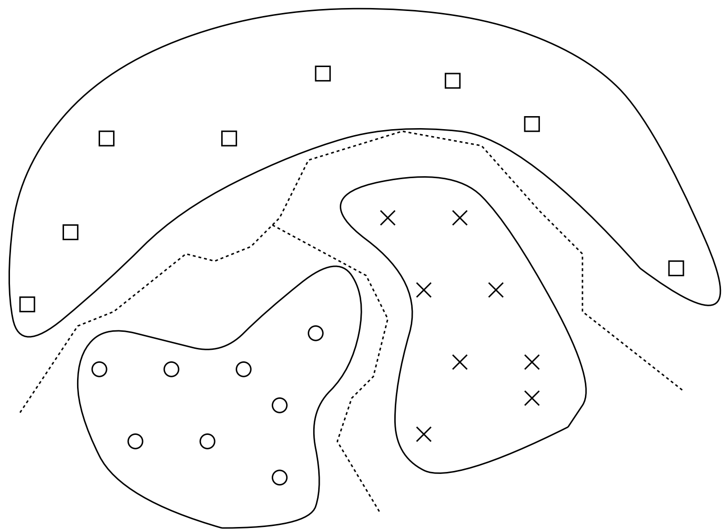
```
N i o:  g1(x)  g2(x)  w1(1)  w1(2)  w2(1)  w2(2)
------------------------------------------------
4 1 1:  1.300  2.100  1.000 -1.500  1.000 -5.500
4 2 2:  4.550  7.950  2.000 -1.700  0.000 -5.300
4 3 2:  3.700  5.300  2.000 -1.700  0.000 -5.300
4 4 1:  1.660 -1.060  2.000 -1.700  0.000 -5.300
4 5 1: -0.040 -6.360  2.000 -1.700  0.000 -5.300
4 6 2:  2.850  2.650  2.000 -1.700  0.000 -5.300
5 1 1:  1.240  2.160  1.000 -1.200  1.000 -5.800
5 2 2:  4.100  8.400  2.000 -1.400  0.000 -5.600
5 3 2:  3.400  5.600  2.000 -1.400  0.000 -5.600
5 4 1:  1.720 -1.120  2.000 -1.400  0.000 -5.600
5 5 1:  0.320 -6.720  2.000 -1.400  0.000 -5.600
5 6 2:  2.700  2.800  2.000 -1.400  0.000 -5.600
6 1 1:  2.280  1.120  2.000 -1.400  0.000 -5.600
```

初期値・増分の影響



区分的線形識別関数

- 線形識別関数により分離可能 $\Longleftrightarrow$ 1 カテゴリあたり 1 プロトタイプの最近傍決定則により識別可能
- カテゴリあたり多数のプロトタイプが必要
- 分離線は線分の集合 (プロトタイプ間の垂直二等分線の集合: **ボロノイ図**)



線形識別関数では分離不可能

区分的線形識別関数

- カテゴリ ω_i の識別関数 $g_i(\boldsymbol{x})$ は L_i 個の線形識別関数 $g_i^{(l)}(\boldsymbol{x})$ によって決まる

$$g_i(\boldsymbol{x}) = \max_{l=1,\ldots,L_i} \{g_i^{(l)}(\boldsymbol{x})\}$$

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

  ただし $i = 1, \ldots, c$

- L_i と w_{ij} の同時学習が必要 $\rightarrow$ 未解決問題

ニューラルネットワーク

- 多層ニューラルネットワーク
- 中間層やユニットの数: L_i(プロトタイプ数) に対応
- 区分線形識別関数と等価
- 学習アルゴリズム: **誤差逆伝播法** (back propagation)
- 数学的基礎: 最小 2 乗法 (least square method), 最急降下法 (steepest descent method), 重回帰分析 (multiple regression analysis)
- 線形代数の基本理論