

最近傍決定則による文字認識

- パターンは 5×5 の25次元2値ベクトル ($d=25$)
- 数字3文字 (3クラス) の認識
- 学習パターンはクラス当り5文字
- それらを平均 (実際は合計) してクラス毎にプロトタイプを求める → プロトタイプが3つできる
- 未知パターンの1つを入力してプロトタイプとのユークリッド距離を求める
- 距離が最小となったものを認識結果として出力

最近傍決定則による文字認識(補足)

- 特徴として各メッシュの黒(文字)、白(背景)情報を利用

黒→1 白→0

- メッシュ数=dとするとd次元の2値特徴ベクトル (d=25)

← d次元 →
特徴ベクトル $X = (1, 0, 0, 1 \cdots 0, 1)$

- プロトタイプPは学習パターンの平均(合計)

- 照合はプロトタイプPとのユークリッド距離

$$|X - P|^2 = \sum_{j=1}^d (x_j - p_j)^2$$

プロトタイプの計算

← 学習パターン →

プロトタイプ
(平均パターン×5)

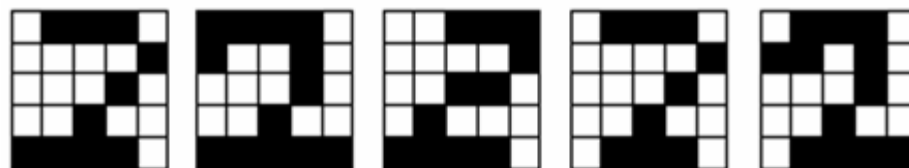
1	4	5	5	1
2	1	0	2	3
0	0	1	5	0
0	1	4	0	0
3	5	5	5	2

(20)

(70)

(90)

クラス「2」



(21)

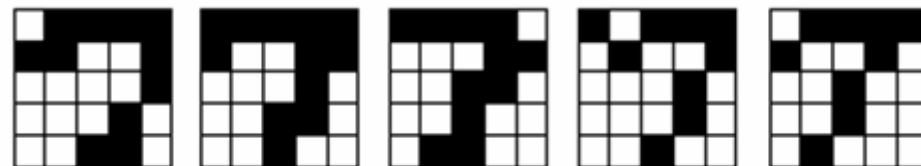
(22)

(23)

(24)

(25)

クラス「7」



(71)

(72)

(73)

(74)

(75)

クラス「9」



(91)

(92)

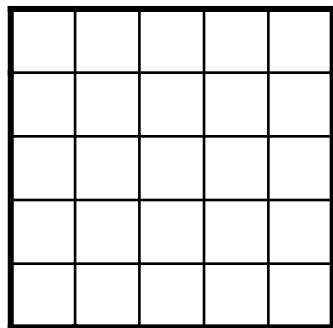
(93)

(94)

(95)

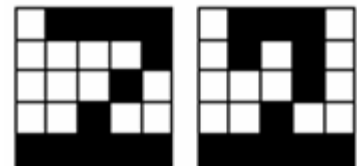
未知パターンとプロトタイプとの距離計算

入力パターン
(未知パターン×5)



未知パターン

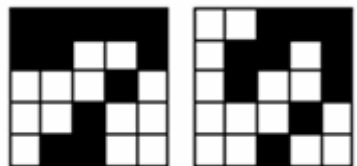
クラス「2」



(26)

(27)

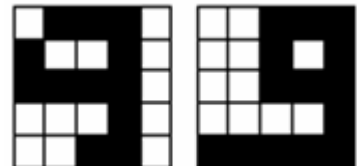
クラス「7」



(76)

(77)

クラス「9」



(96)

(97)

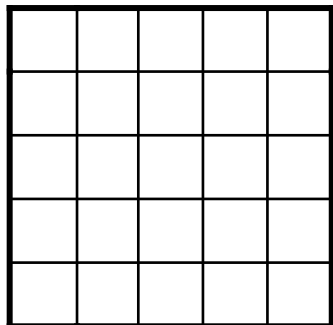
プロトタイプ

クラス「2」

1	4	5	5	1
2	1	0	2	3
0	0	1	5	0
0	1	4	0	0
3	5	5	5	2

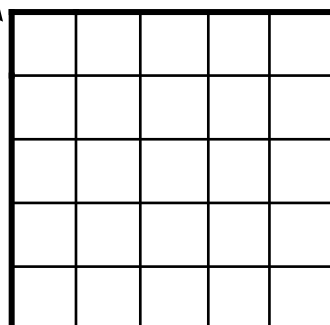
(20)

()
クラス「7」



(70)

クラス「9」



(90)

各メッシュ間の差の二乗

距離 =

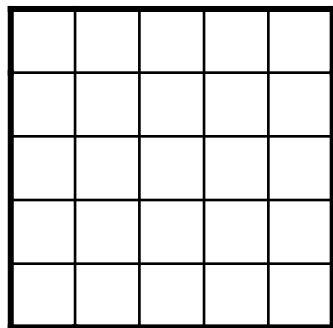
距離 =

距離 =

認識結果 =

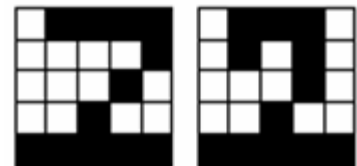
未知パターンとプロトタイプとの距離計算

入力パターン
(未知パターン×5)



未知パターン

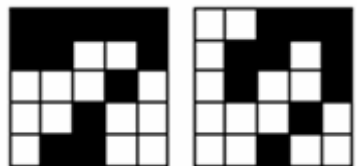
クラス「2」



(26)

(27)

クラス「7」



(76)

(77)

クラス「9」



(96)

(97)

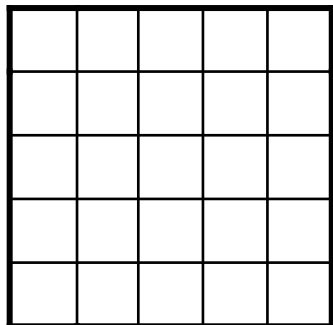
プロトタイプ

クラス「2」

1	4	5	5	1
2	1	0	2	3
0	0	1	5	0
0	1	4	0	0
3	5	5	5	2

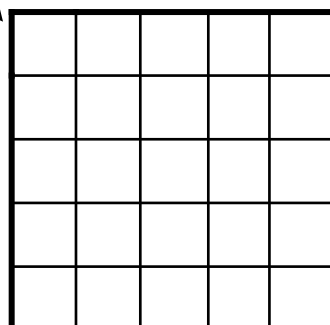
(20)

()
クラス「7」



(70)

クラス「9」



(90)

各メッシュ間の差の二乗

距離 =

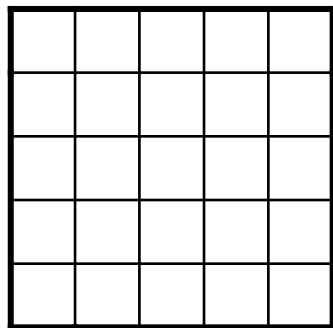
距離 =

距離 =

認識結果 =

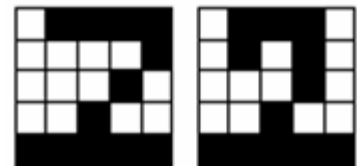
未知パターンとプロトタイプとの距離計算

入力パターン
(未知パターン×5)



未知パターン

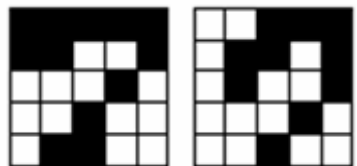
クラス「2」



(26)

(27)

クラス「7」



(76)

(77)

クラス「9」



(96)

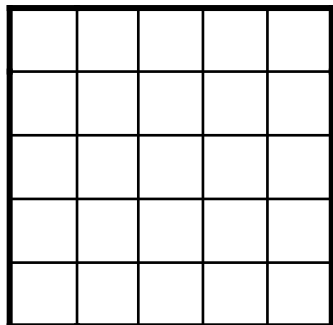
(97)

クラス「2」

1	4	5	5	1
2	1	0	2	3
0	0	1	5	0
0	1	4	0	0
3	5	5	5	2

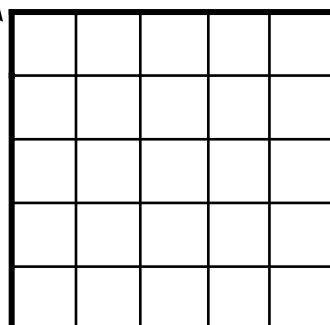
(20)

()
クラス「7」



(70)

クラス「9」



(90)

プロトタイプ

各メッシュ間の差の二乗

距離 =

距離 =

距離 =

認識結果 =