

機械学習 最近傍法

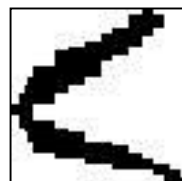
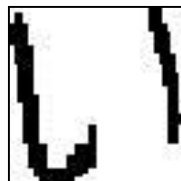
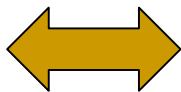
管理工学科
篠沢佳久

最近傍法

文字認識

文字認識

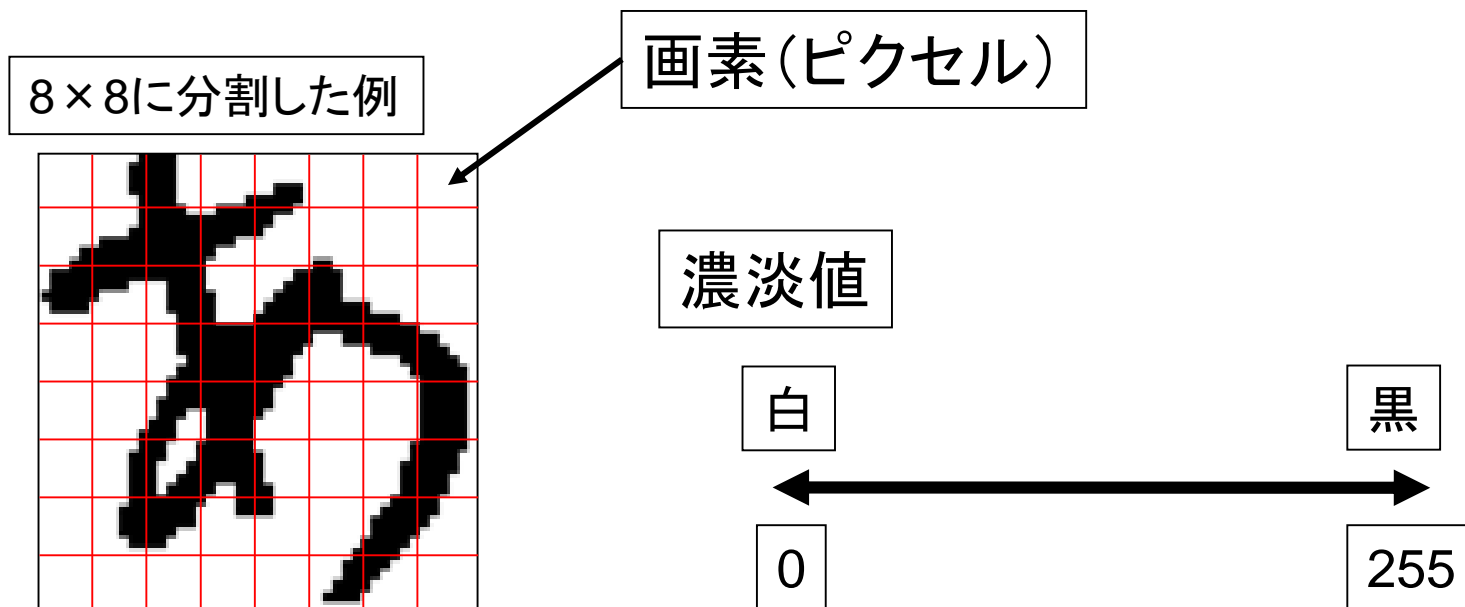
どれと一致しますか



産業技術総合研究所 文字画像データベースETL9B

標本化と量子化処理

- 文字画像を($X \times Y$)個に均等に分割(標本化処理)

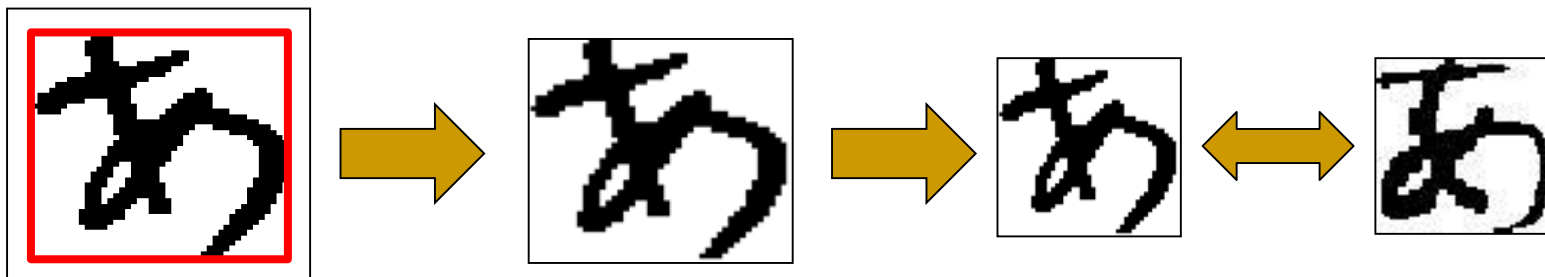


- 各画素において、白と黒の割合(濃淡)を(例えば)256段階(0～255の値)で示す(量子化処理)

*メッシュとも呼びます

前処理

- 文字(と考えられる)領域を切り出す
- 文字の大きさを一定になるように変換
 - 例えばアフィン変換を用いる
- これを前処理(もしくは正規化処理)と呼ぶ



切り出し

比較したい文字画像と同じ大きさにする(この場合は縮小)

特徴ベクトル①

- 文字の濃淡値は 8×8 の行列で表現できる

$$X = \begin{pmatrix} x_{11} & x_{21} & \cdots & x_{81} \\ x_{12} & x_{22} & & x_{82} \\ \vdots & & \ddots & \vdots \\ x_{18} & x_{28} & \cdots & x_{88} \end{pmatrix}$$

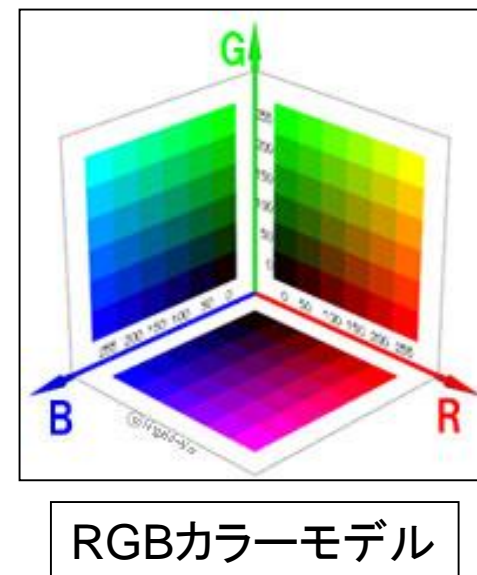
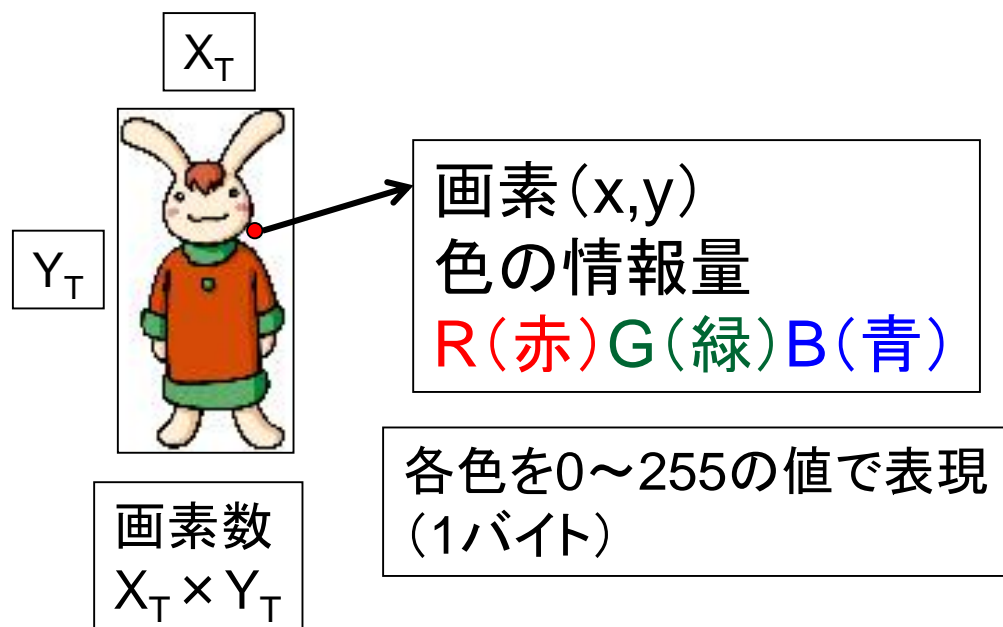
x_{ij} : 画素(i,j)の濃淡値
0~255の値

- しかし、一般的にはベクトルにて表現
 - この場合は64次元
 - これを**特徴量**もしくは**特徴ベクトル**と呼ぶ

$$\mathbf{x}^t = (x_1, x_2, \cdots, x_{64})$$

カラー画像の場合①

- ビットマップ形式, ラスターイメージ
- 画素単位に情報量を持つ



RGBカラーモデル

【描画ツール→図形の塗りつぶし/ 図形の枠線→その他の色→ユーザー指定】



RGBカラーモデル

各パラメータ値は(0～255)で指定

カラー画像の場合

■ 大きさが8×8の場合

$$X = \begin{pmatrix} x_{11} & x_{21} & \cdots & x_{81} \\ x_{12} & x_{22} & & x_{82} \\ \vdots & & \ddots & \vdots \\ x_{18} & x_{28} & \cdots & x_{88} \end{pmatrix}$$

$x_{ij} : (r_{ij}, g_{ij}, b_{ij})$
 r_{ij} : 赤, g_{ij} : 緑, b_{ij} : 青
それぞれ0~255の値

■ ベクトルで表記した場合

$$\mathbf{x}^t = (x_1, x_2, \cdots, x_{64}) \quad \mathbf{x}_i : (r_i, g_i, b_i)$$

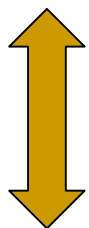


$$\mathbf{x}^t = (x_1, x_2, \cdots, x_{192})$$

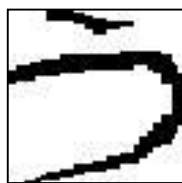
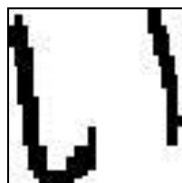
特徴ベクトル②



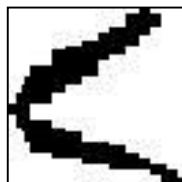
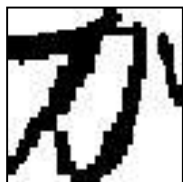
調べたい文字の特徴ベクトル \mathbf{t}



ベクトル \mathbf{t} と10個のベクトル \mathbf{x}_p の「類似度」をそれぞれ計算し、最も「類似度」の高い文字を認識結果とする



比較したい文字の
特徴ベクトル



$\mathbf{x}_p (p = 1, 2, \dots, 10)$

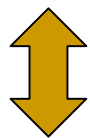
二つのベクトルの類似度を求めるには？

特徴ベクトル③



$$\mathbf{t} = (3, 14, 2, \dots, 12, 2)^t$$

64次元

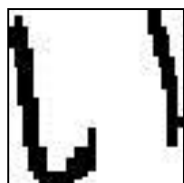


$$\mathbf{x}_1 = (5, 11, 4, \dots, 15, 3)^t$$

64次元



$$\mathbf{x}_4 = (1, 8, 10, \dots, 17, 9)^t$$



$$\mathbf{x}_2 = (16, 10, 9, \dots, 6, 0)^t$$

⋮



$$\mathbf{x}_3 = (1, 1, 2, \dots, 10, 0)^t$$



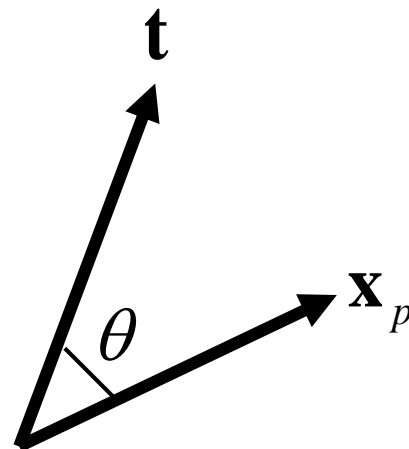
$$\mathbf{x}_{10} = (5, 11, 10, \dots, 12, 9)^t$$

類似度①

■ 類似度

- 二つのベクトルが一致する場合は $\theta = 0$
- 従って R_p は1となる

$$R_p = \cos \theta = \frac{\mathbf{t}^t \mathbf{x}_p}{\|\mathbf{t}\| \cdot \|\mathbf{x}_p\|} = \frac{\sum_{i=1}^{64} t_i x_{pi}}{\sqrt{\sum_{i=1}^{64} t_i^2} \sqrt{\sum_{i=1}^{64} x_{pi}^2}}$$



- R_p が最も1に近い文字 p を認識結果とする

類似度②

■ 相互相関係数

□ 特徴ベクトルが n次元の場合

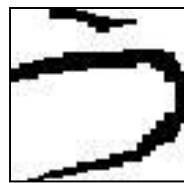
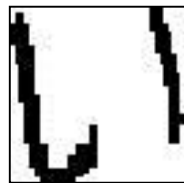
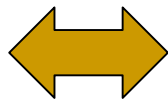
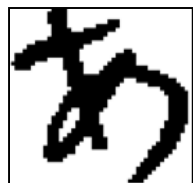
$$R'_p = \cos \theta = \frac{\sum_{i=1}^n (t_i - \bar{t})(x_{pi} - \overline{x_p})}{\sqrt{\sum_{i=1}^n (t_i - \bar{t})^2} \sqrt{\sum_{i=1}^n (x_{pi} - \overline{x_p})^2}}$$

平均値

$$\bar{t} = \frac{1}{n} \sum_{i=1}^n t_i$$

$$\overline{x_p} = \frac{1}{n} \sum_{i=1}^n x_{pi}$$

相互相関係数による類似度



相互相関係数
を求めると...

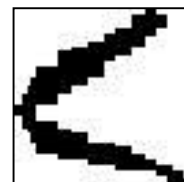
0.902

0.231

0.554

0.612

0.794



0.651

0.428

0.415

0.275

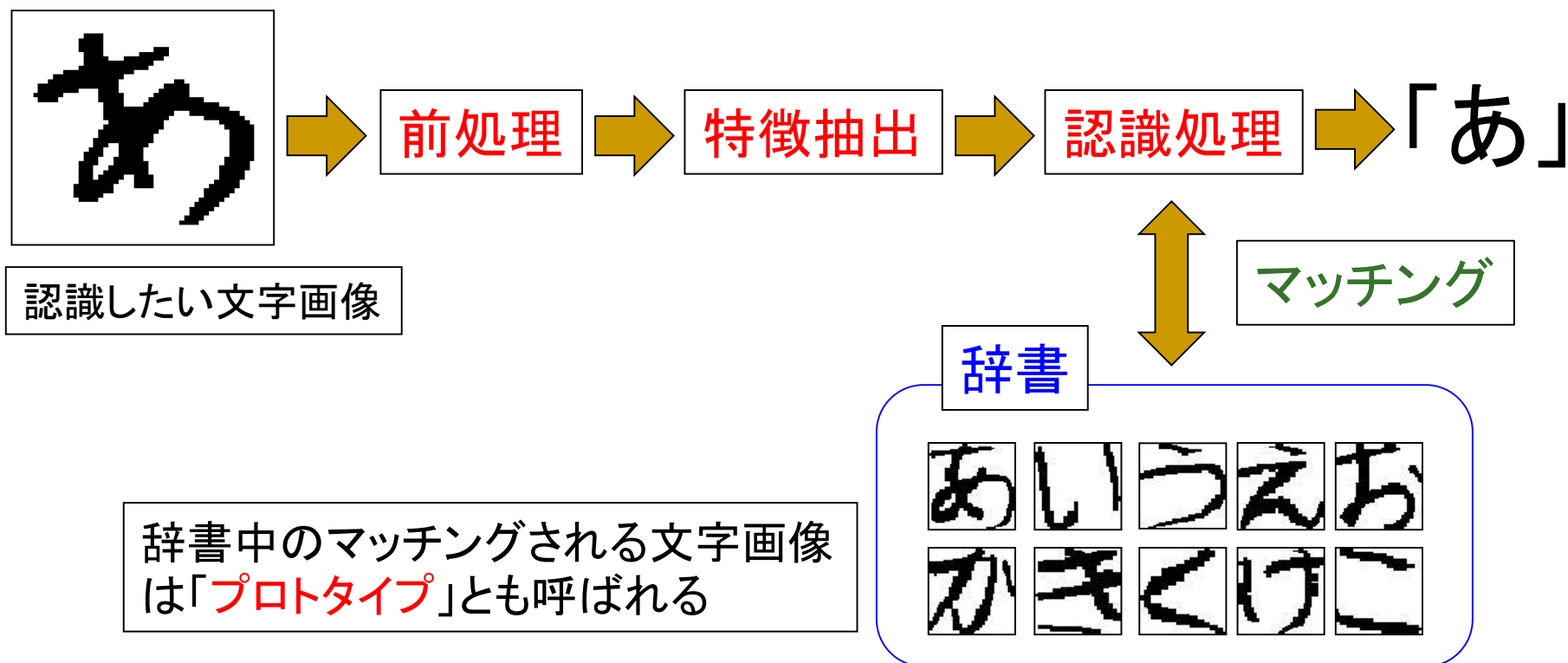
0.327



認識結果



文字認識の基本的な流れ



辞書内のプロトタイプとマッチングを行ない、最も類似度の高いものを結果とする方法を**最近傍法 (Nearest Neighbor)**と呼ぶ

最近傍法のアルゴリズム

$\text{max} = -\infty$

前処理(認識したい文字画像)

\mathbf{t} = 特徴抽出(前処理後の文字画像)

for p in range(Max_p):

$\text{similarity} = \text{類似度}(\mathbf{t}, \mathbf{x}_p)$

 if $\text{similarity} > \text{max}$:

$\text{max} = \text{similarity}$

$\text{answer} = p$

Max_p : プロトタイプの総数

\mathbf{t} : 認識したい文字画像の特徴ベクトル

\mathbf{x}_p : プロトタイプ p の特徴ベクトル

プロトタイプ answer が認識結果

類似性の求め方

類似度

距離

類似度

■ 相互相関係数

□ R_p (R'_p) が1に近いほど, 類似しているものと判断

$$R_p = \cos \theta = \frac{\mathbf{x}^t \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$$

$$R'_p = \cos \theta = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

距離による類似性の計算

- 二つのベクトル間 (x と y) の類似度

n次元ベクトル

- 各要素の差の合計 (距離)

$$\sum_{i=1}^n |x_i - y_i|$$

0に近いほど類似している

距離尺度①

■ ユークリッド距離

$$\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

■ べき乗距離 (ミンコスキー距離)

$$\left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}$$

p=r=1の場合, マンハッタン距離
p=r=2の場合, ユークリッド距離

距離尺度②

■ チェビシェフ距離

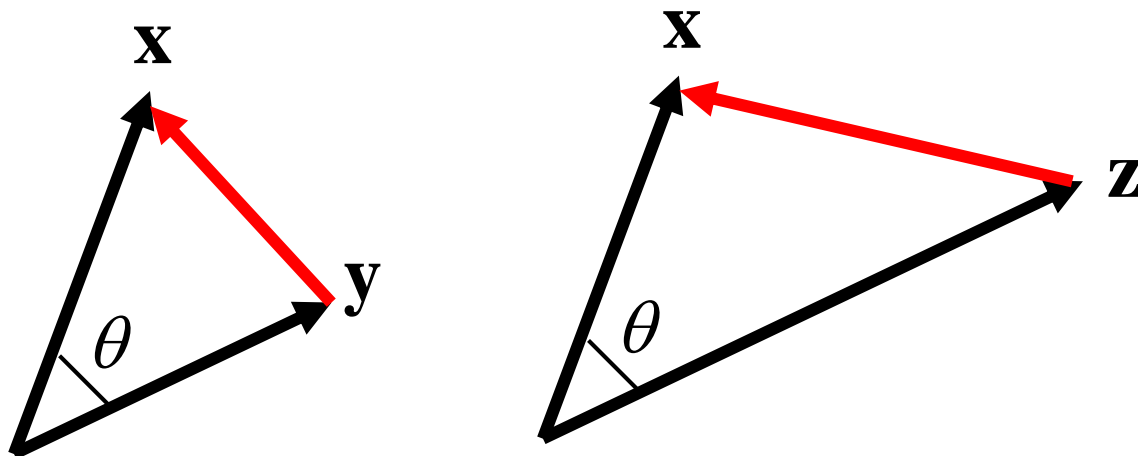
$$\max_{i=1,2,\dots,n} |x_i - y_i|$$

各特徴間要素の最大値を距離とする

■ マハラノビス距離

- 分布を考慮
- 次回以降に説明します

類似度と距離による違い①



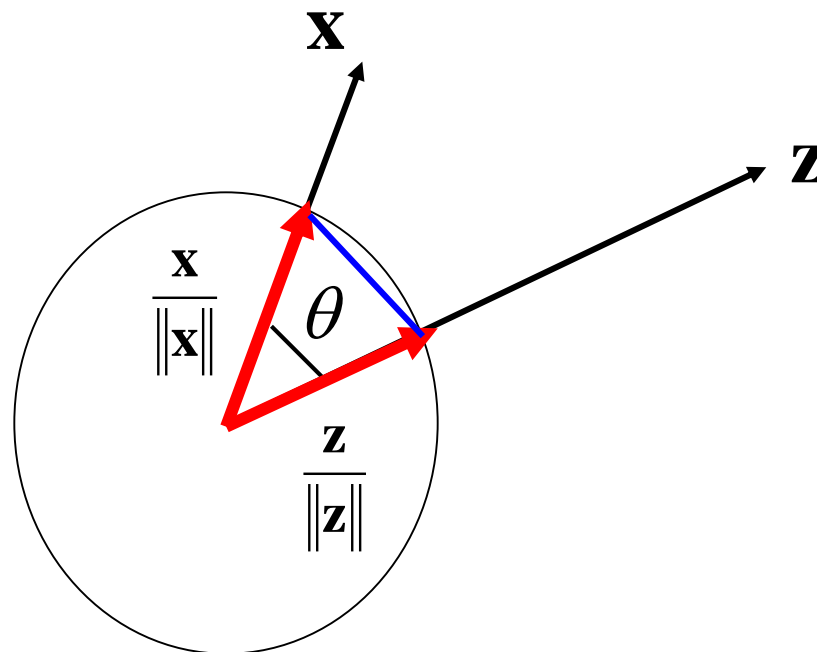
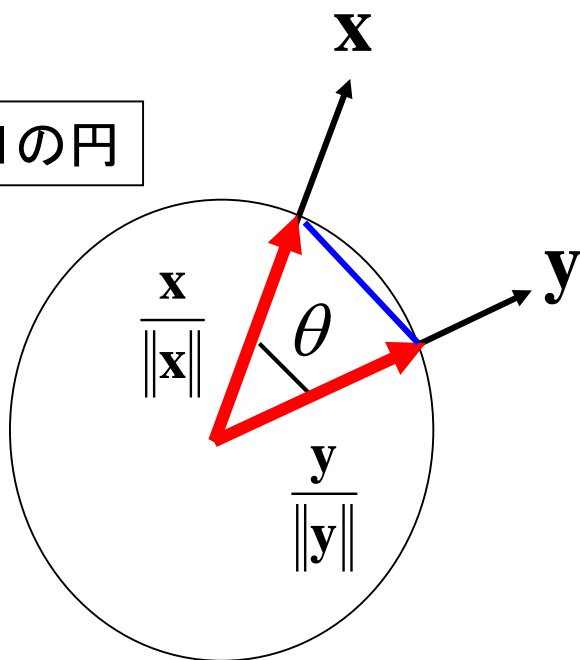
ベクトルによっては類似度は同じであるが、距離は異なるという場合が生じる

→ 類似度を用いた場合と距離を用いた場合では認識結果が異なる場合もある

類似度と距離による違い②

ベクトルのノルム(原点とのユークリッド距離)を1に正規化
→ 二つのベクトル間の距離を求める

半径1の円



特徴ベクトルによる類似性

■ 二つの文字画像の類似性

- 認識したい文字画像, 辞書中の文字画像(プロトタイプ)は特徴ベクトルによって表現される
- 文字画像の類似性は特徴ベクトルの類似度によって求めることができる

■ 認識結果

- 類似度の場合 → 最大となるプロトタイプ
- 距離の場合 → 最小となるプロトタイプ

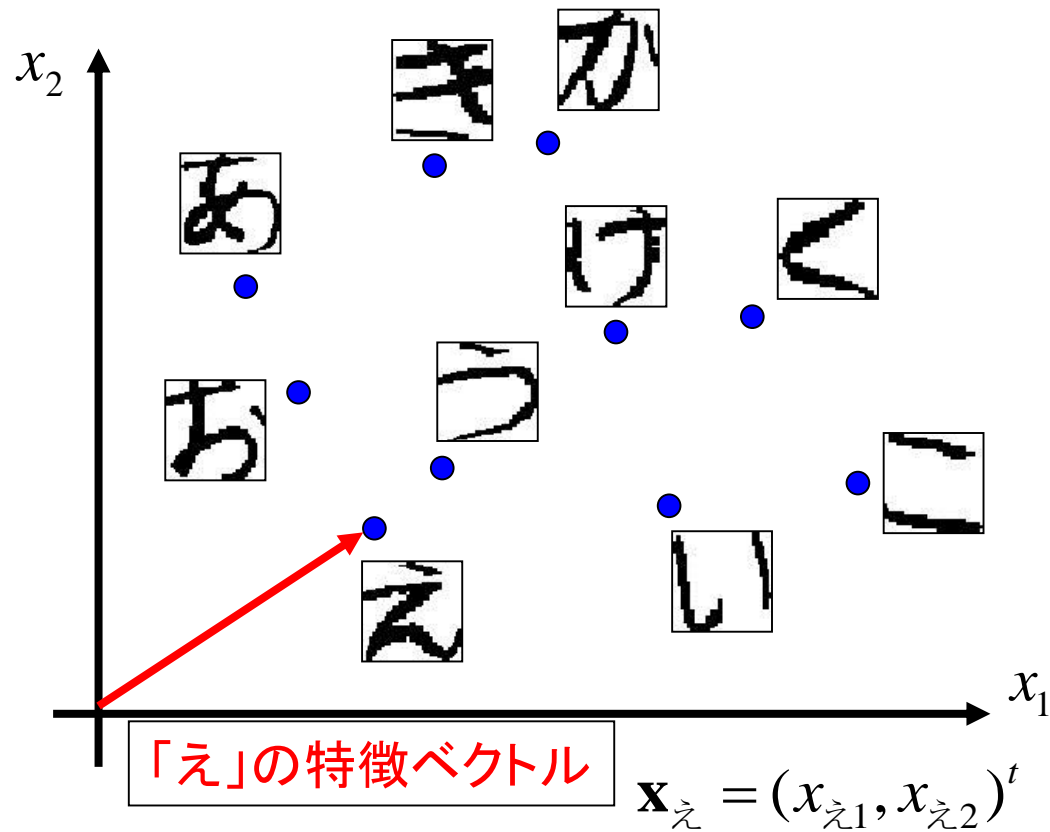
特徴空間①

■ 特徴空間

- 特徴ベクトルによって張られる空間
- 各文字画像の特徴ベクトルは、特徴空間上の一点としても表現できる
- 特徴空間上では、特徴ベクトルが類似している文字画像は近くに、類似していない文字画像は遠くに配置される
 - 塊(クラスター)ができる
 - 必ずしも同種の文字によってクラスターが生成されるとは限らない

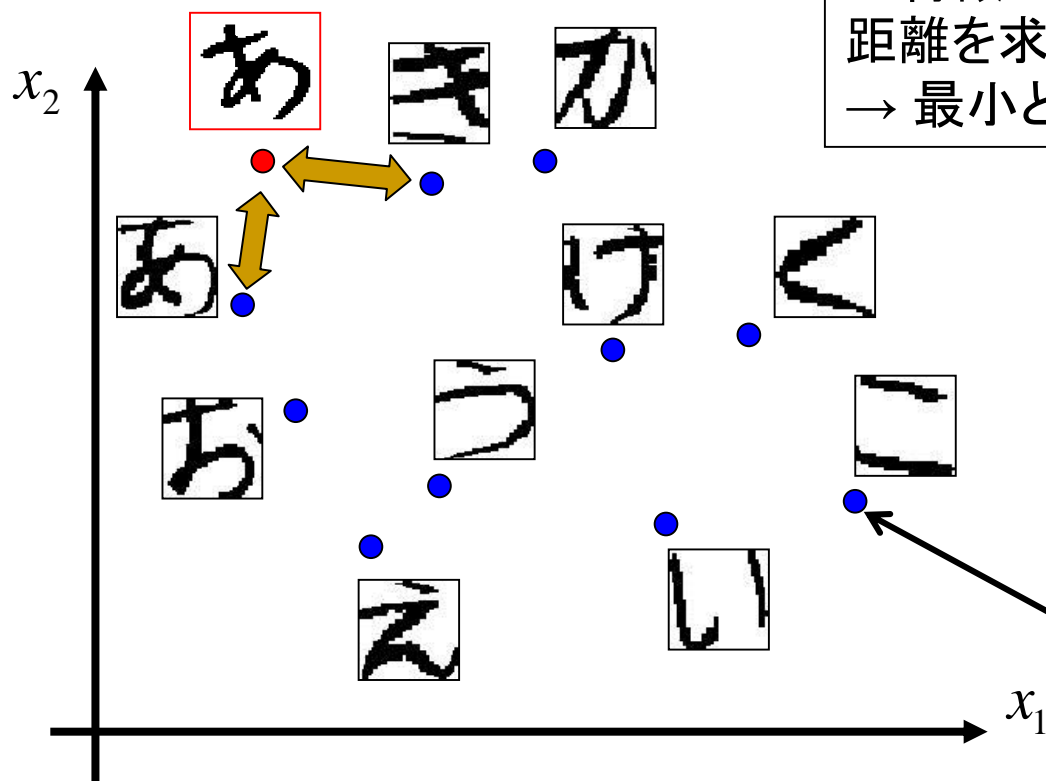
特徴空間②

- 仮に二次元上で表現できたとしたら...



最近傍法

認識したい文字画像
(未知の文字画像)



特徴空間上において、認識したい文字画像と辞書の全ての文字画像(プロトタイプ)の特徴ベクトル(特徴空間上の座標)との距離を求める
→ 最小となるものを認識結果*

最近傍法
(Nearest Neighbor)

プロトタイプ

*類似度の場合は最大となるものを認識結果とします

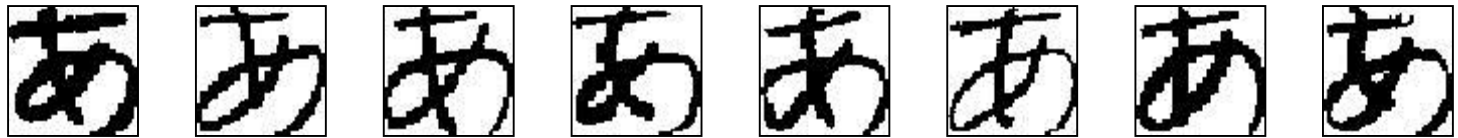
最近傍法の問題点

- (例えば)「あ」という文字
 - 書き手によって千差万別
 - 辞書中のプロトタイプ「あ」によって認識結果が異なる場合もある
- 辞書中のプロトタイプ「あ」はどのように作成すればよいか？
 - 複数個のプロトタイプを利用
 - 最近傍法を改良(k近傍法)
 - 出現確率(分布)を考慮 → 統計的機械学習

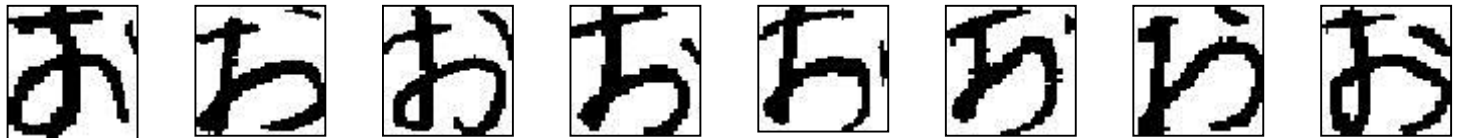
k 近傍法

複数個のプロトタイプ

■「あ」の場合

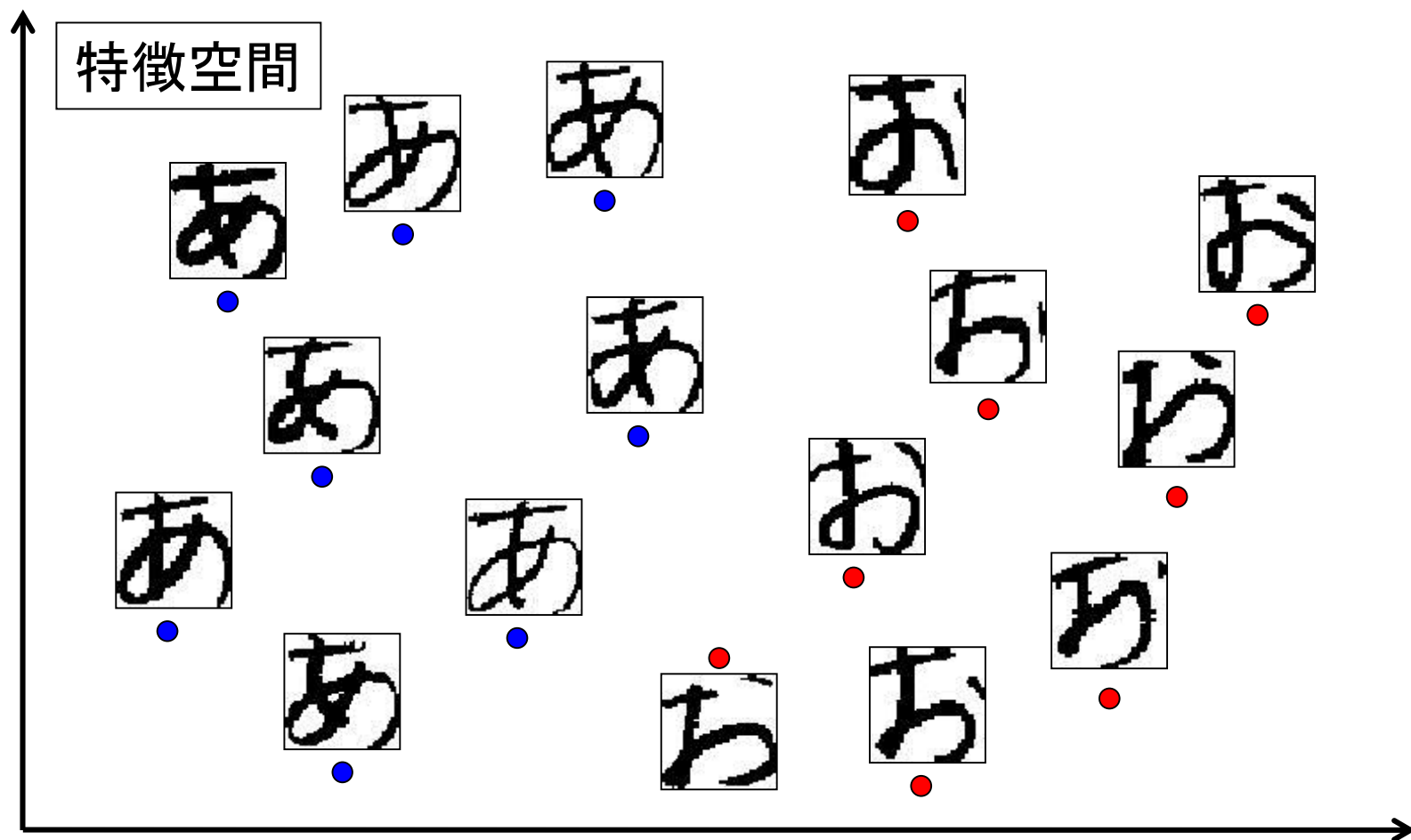


■「お」の場合

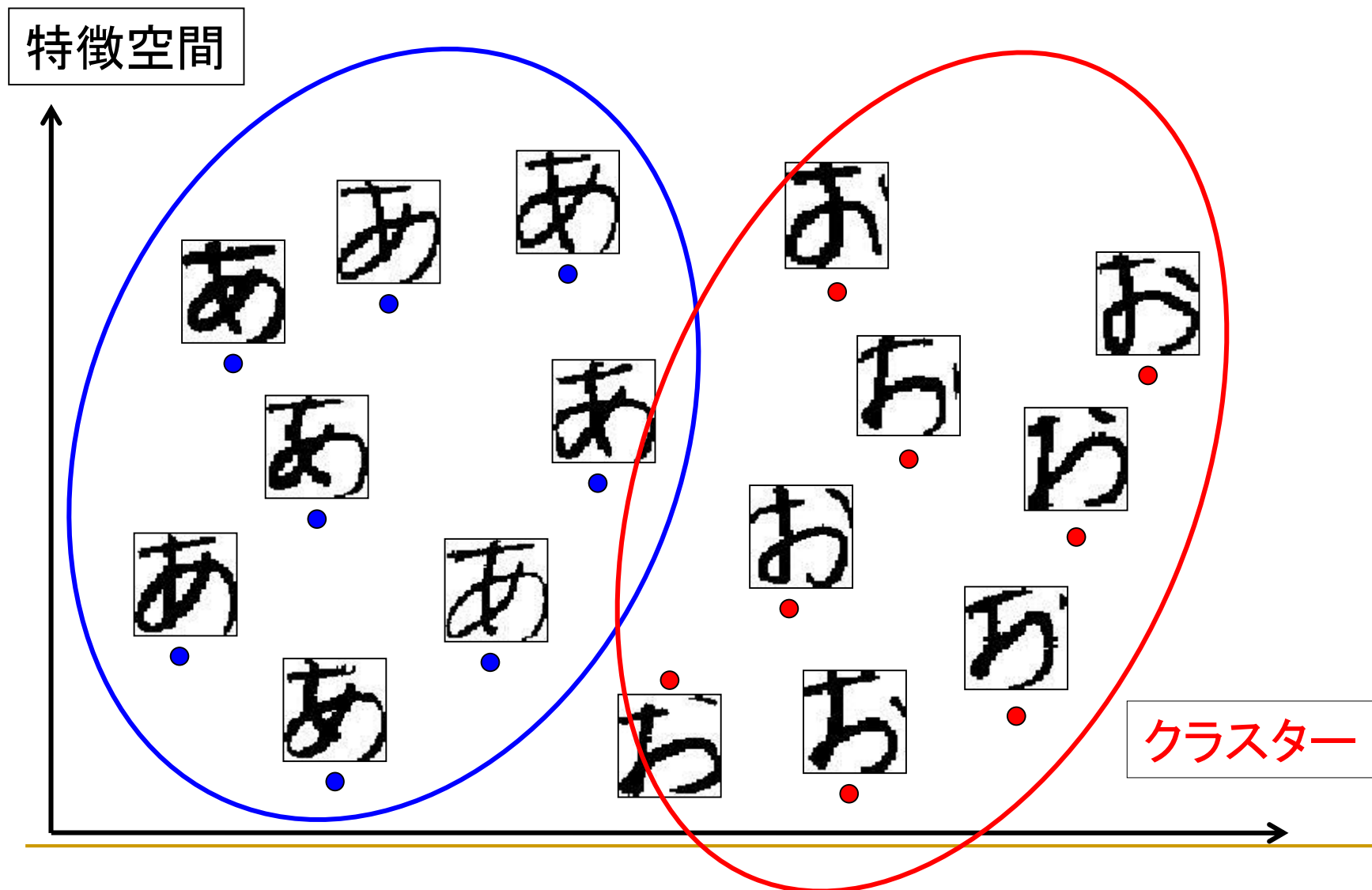


最近傍法の改良(k 近傍法)①

一つの文字画像について複数個のプロトタイプを用意

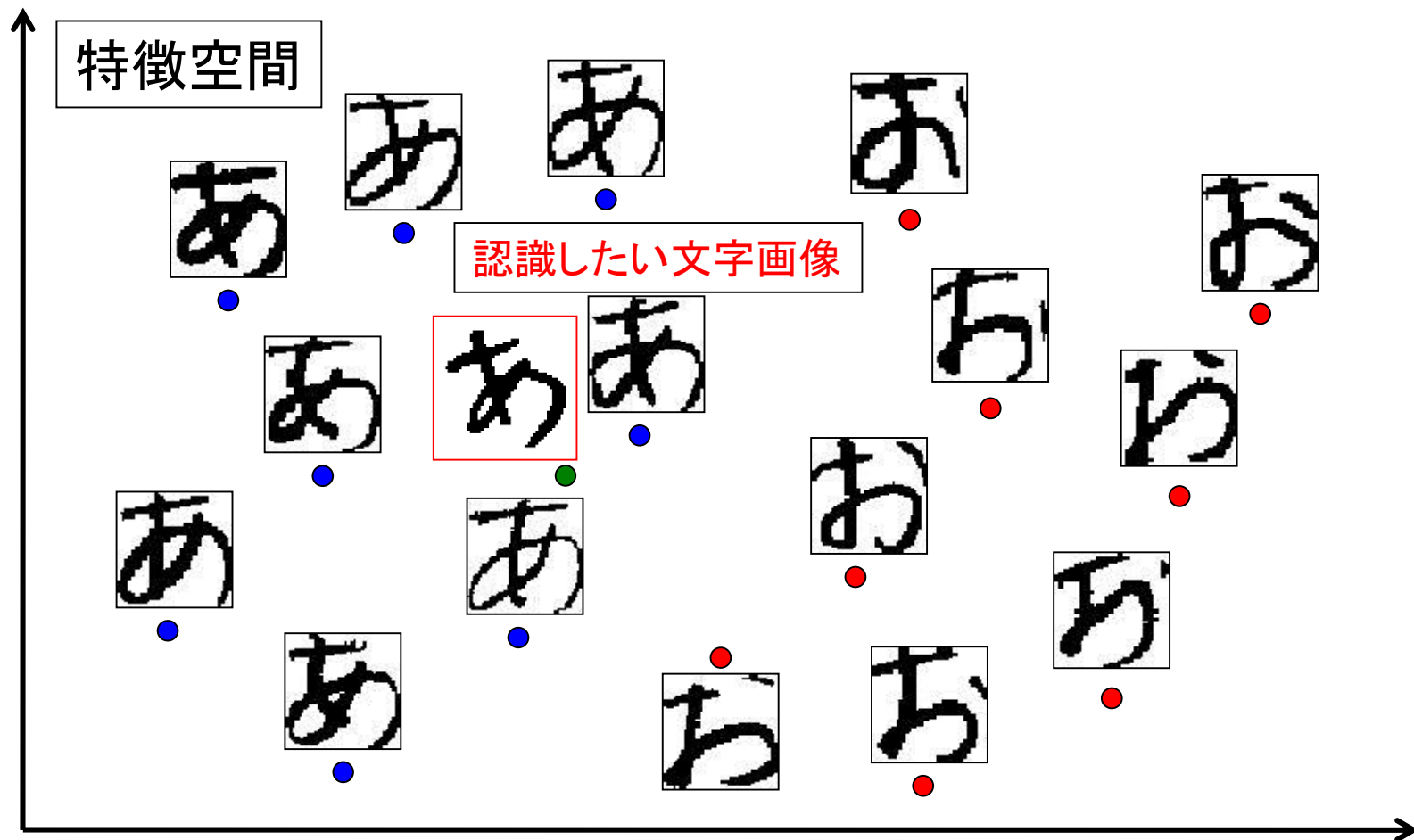


特徴空間上でのパターンの分布



最近傍法の改良(k 近傍法)②

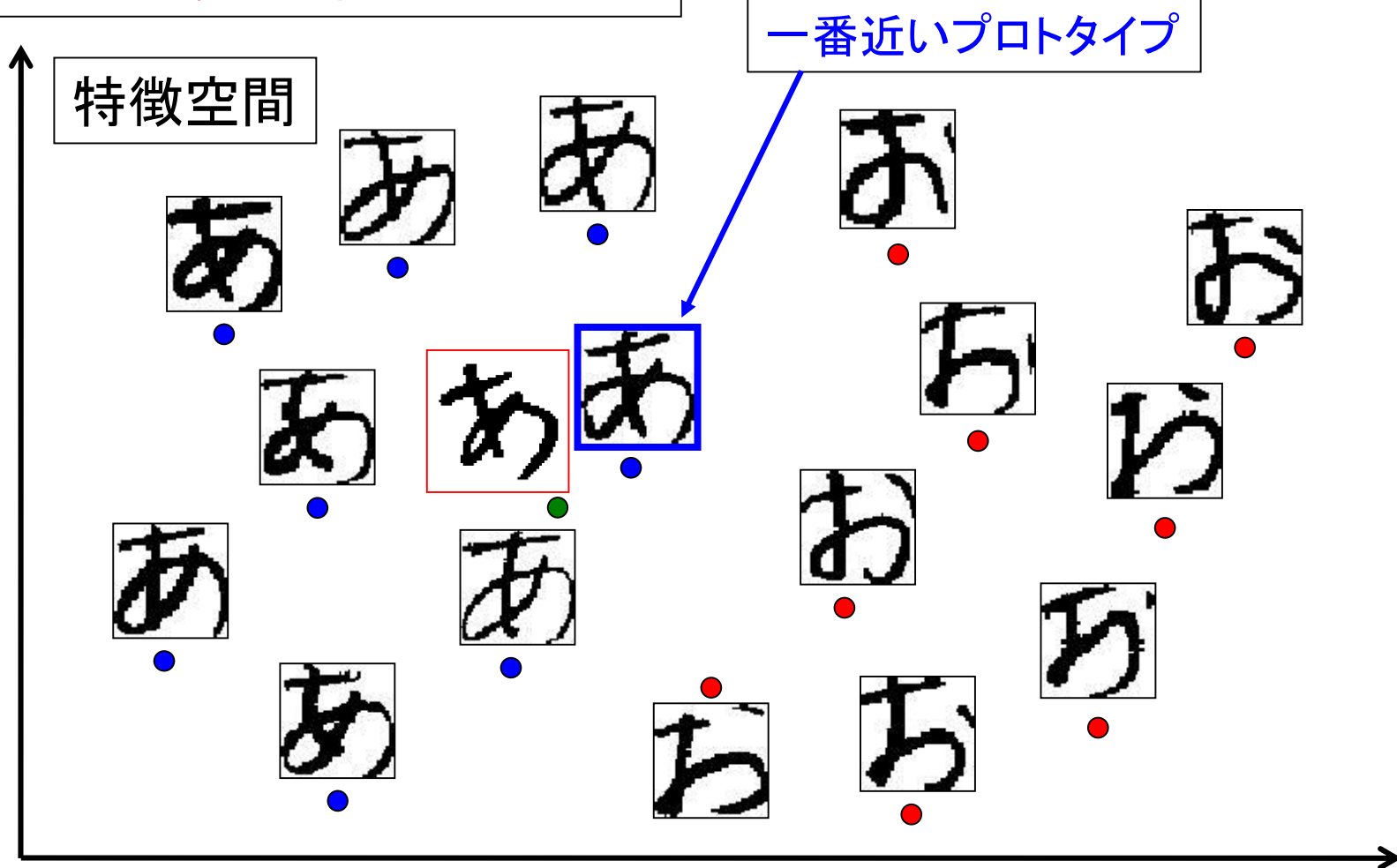
認識したい文字画像「あ」を特徴空間上に配置



k近傍法③

一番目まで近いプロトタイプは「あ」
→「あ」と認識

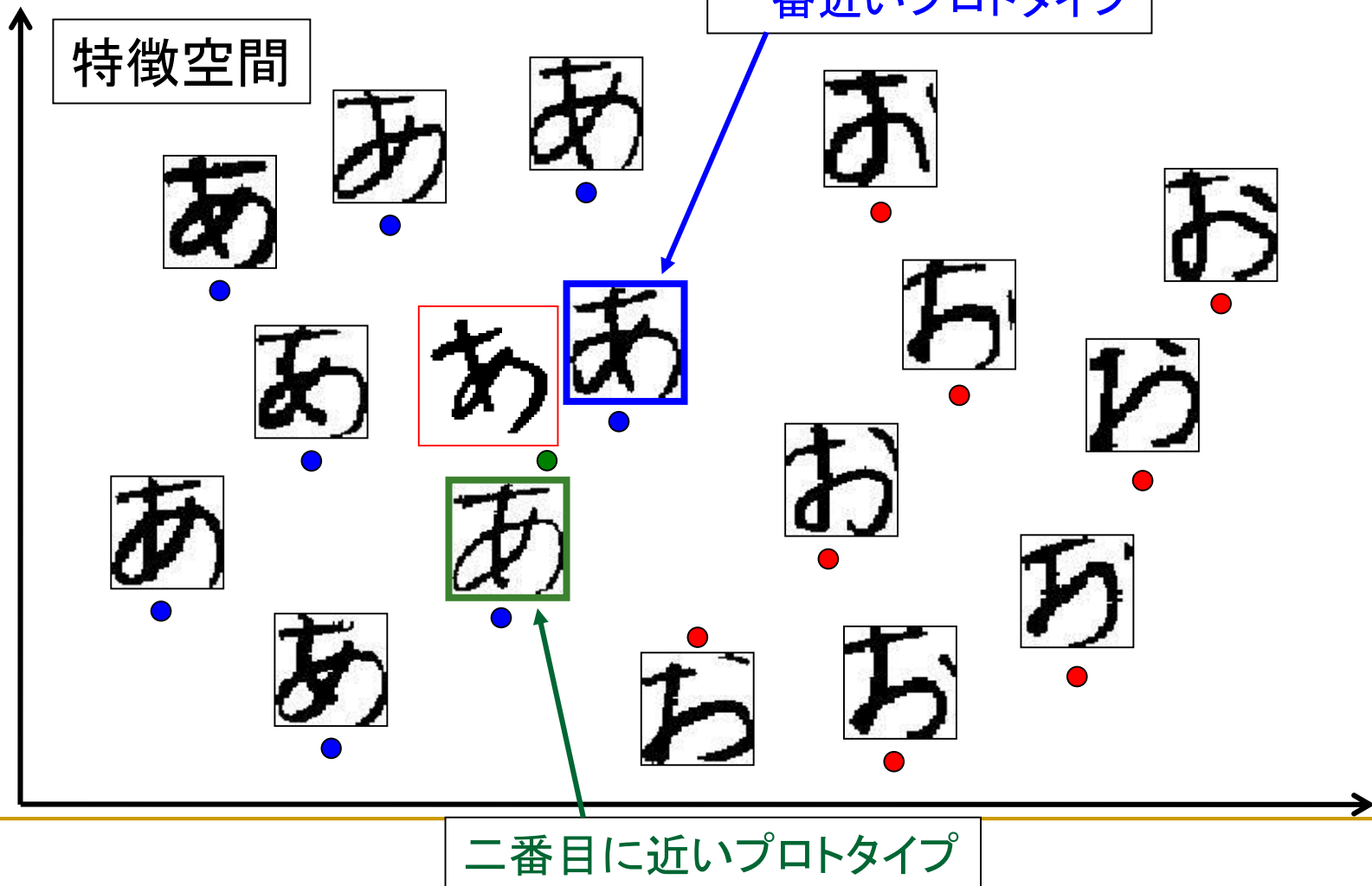
k=1 の場合(最近傍と同じ)



k近傍法④

二番目までに近いプロトタイプは「あ」が2個
→「あ」と認識

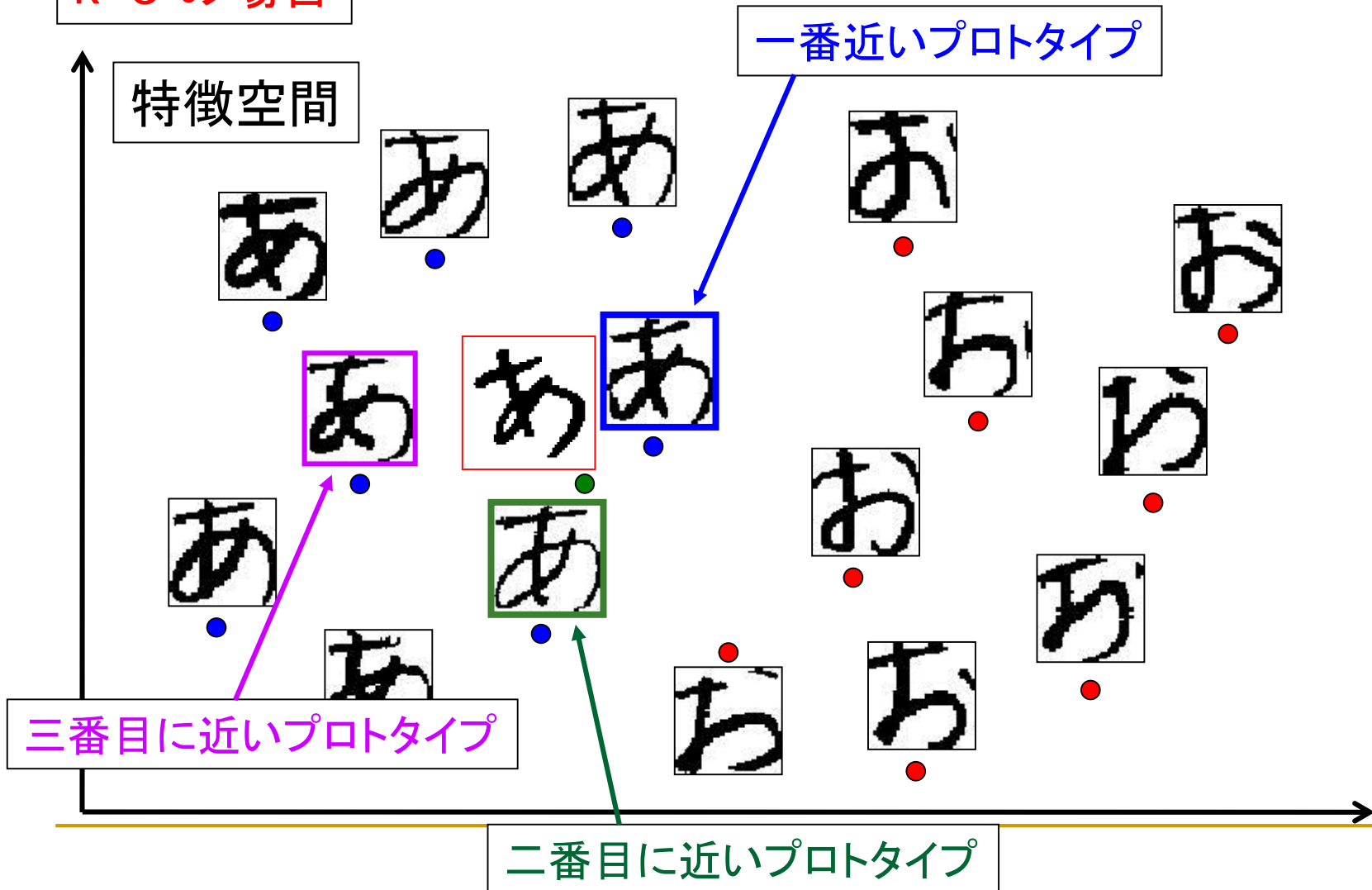
k=2 の場合



k近傍法⑤

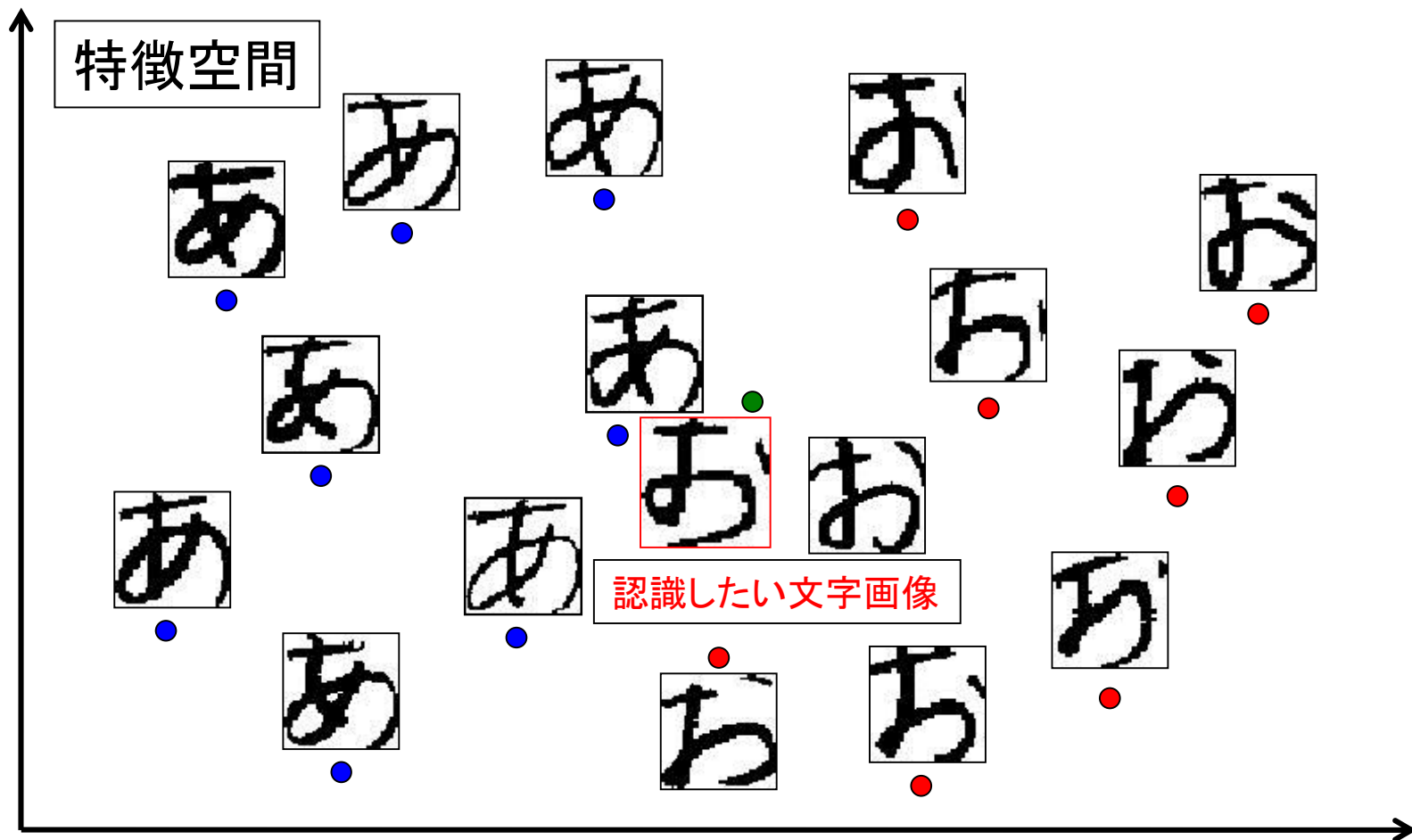
三番目までに近いプロトタイプは「あ」が3個
→「あ」と認識

k=3 の場合



k近傍法⑥

認識したい文字画像「お」を特徴空間上に配置

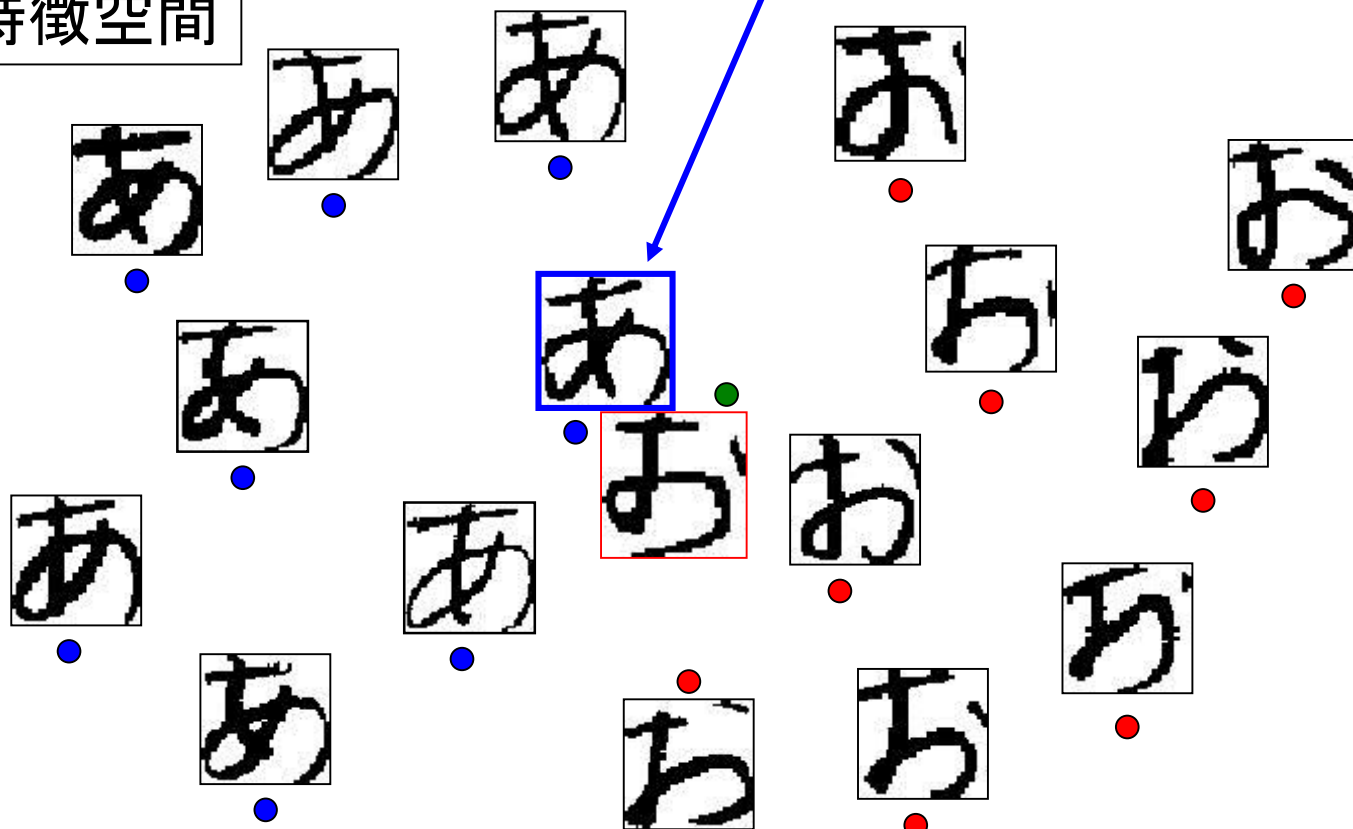


k近傍法⑦

k=1 の場合

特徴空間

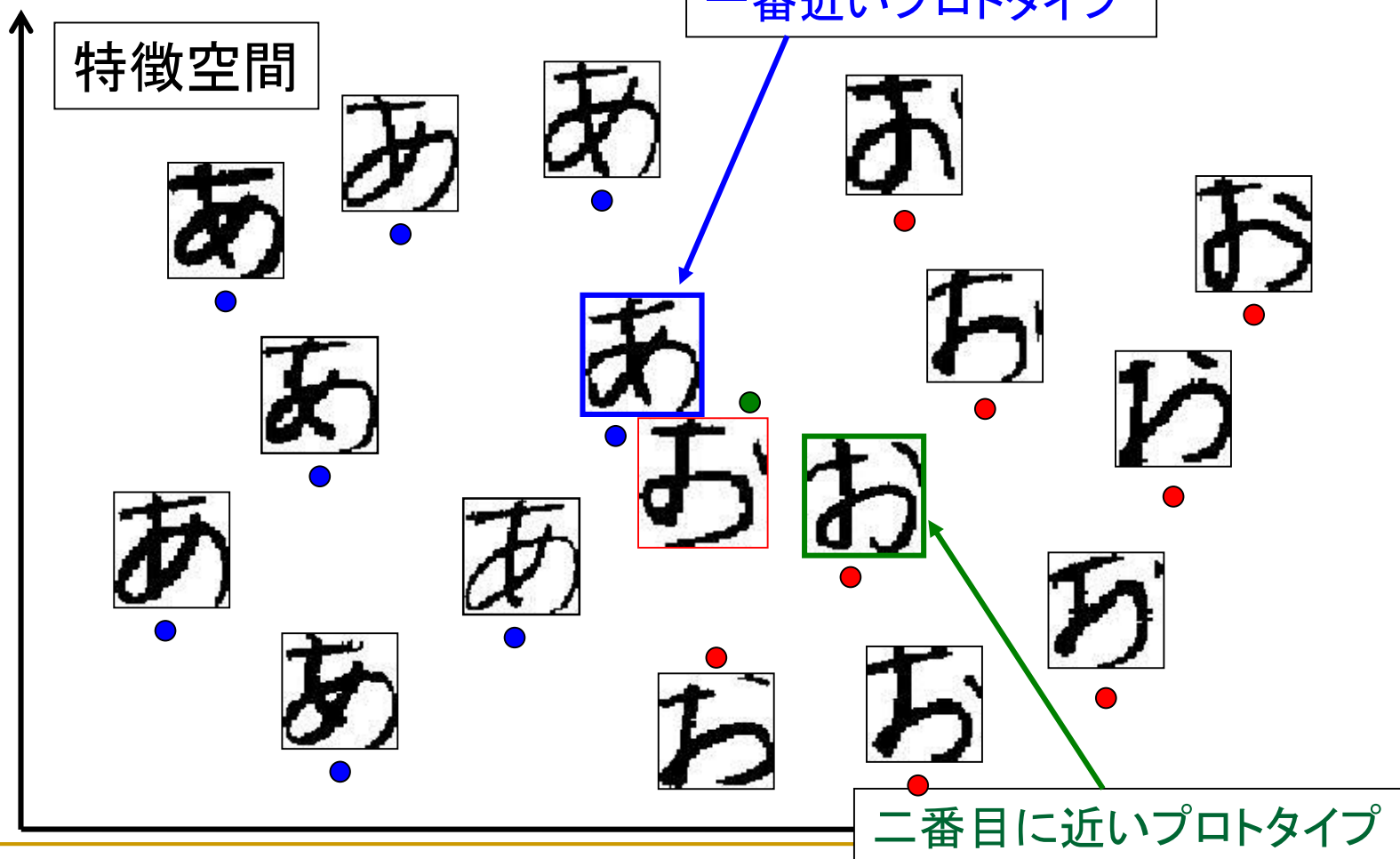
一番近いプロトタイプは「あ」
→「あ」と認識される(誤認識)



k近傍法⑧

二番目まで近いプロトタイプを調べると「あ」が1個,「お」が1個
→確定できない

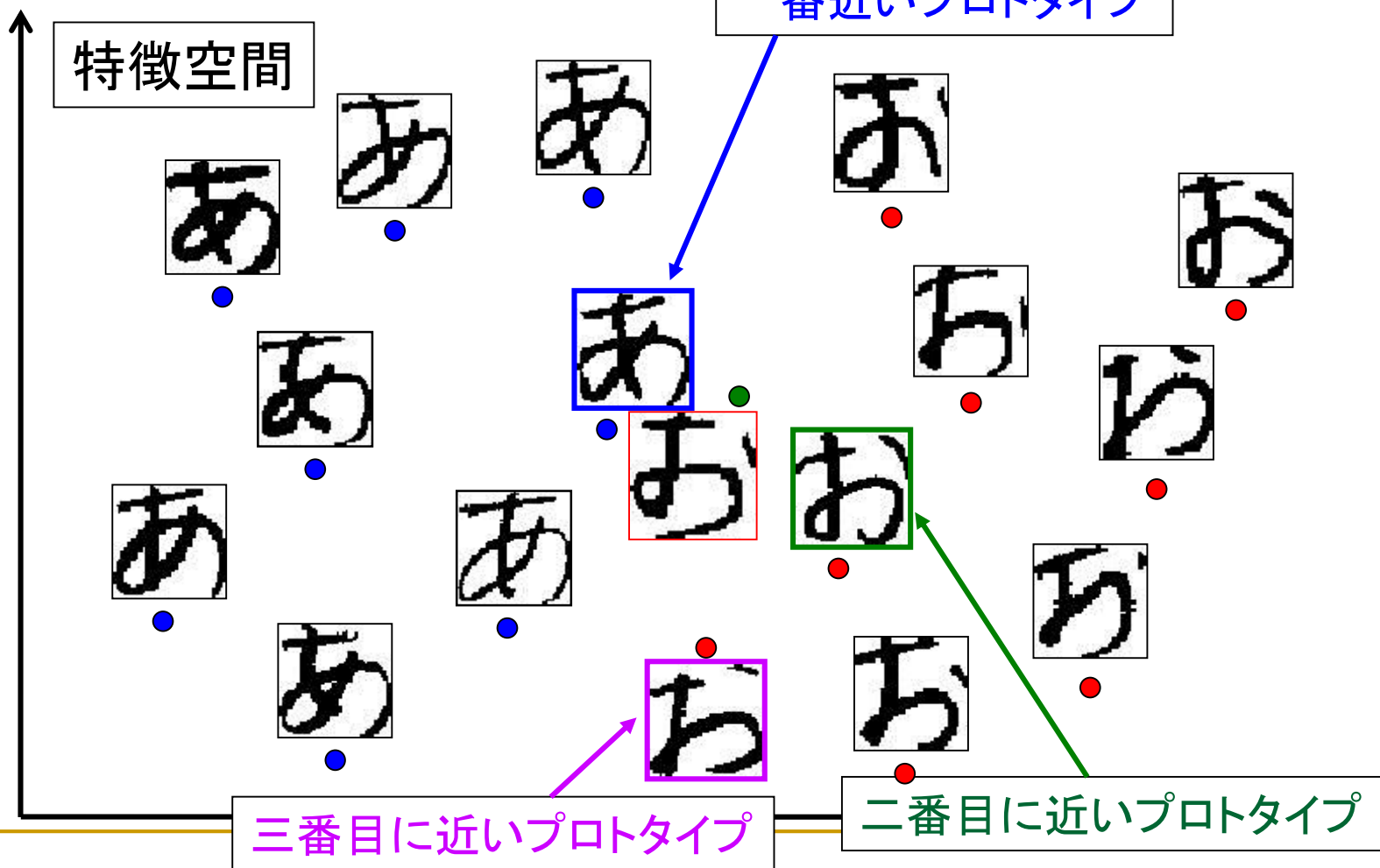
k=2 の場合



k近傍法⑨

三番目まで近いプロトタイプを調べると「あ」が1個,「お」が2個
→「お」と認識

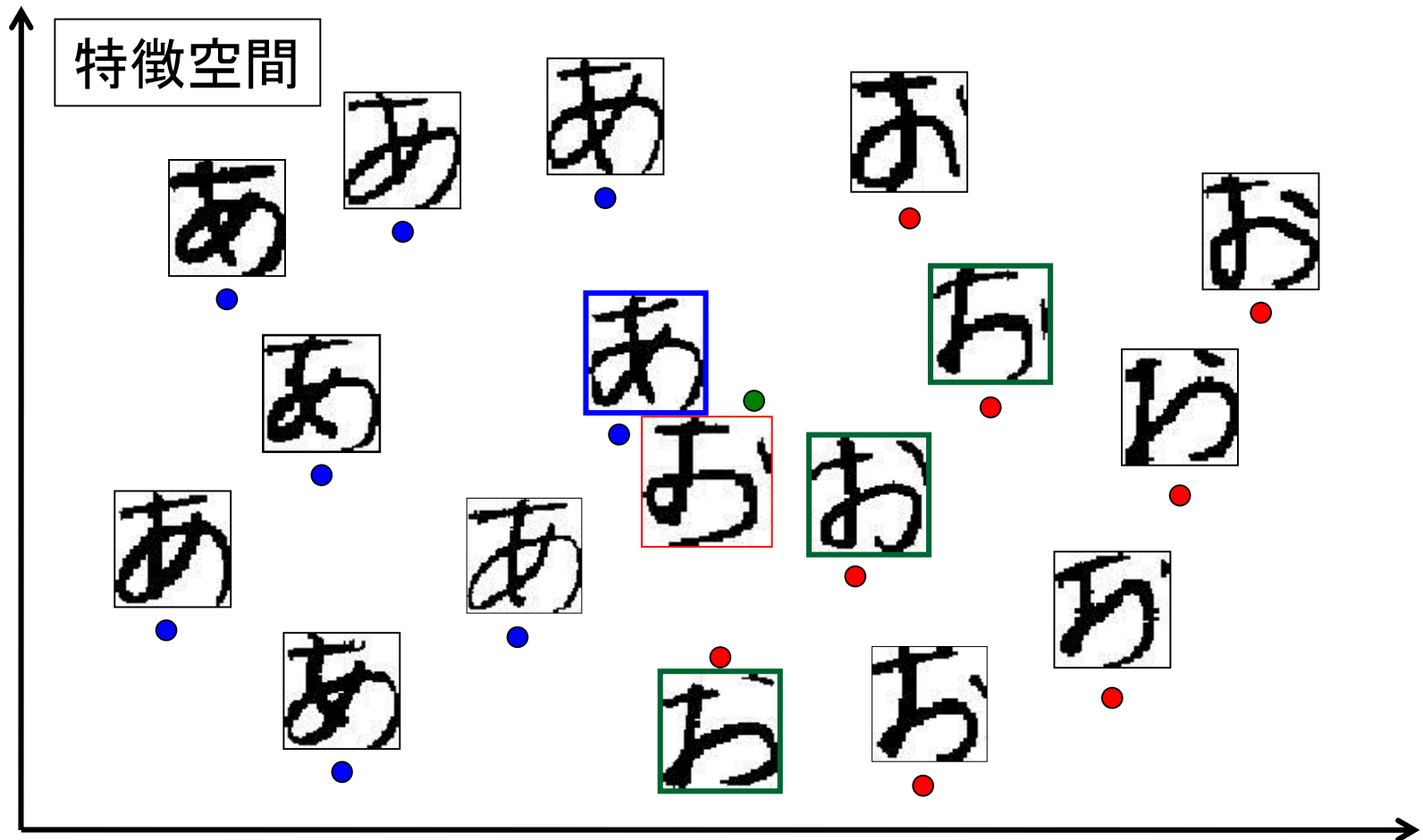
k=3 の場合



k近傍法⑩

k=4 の場合

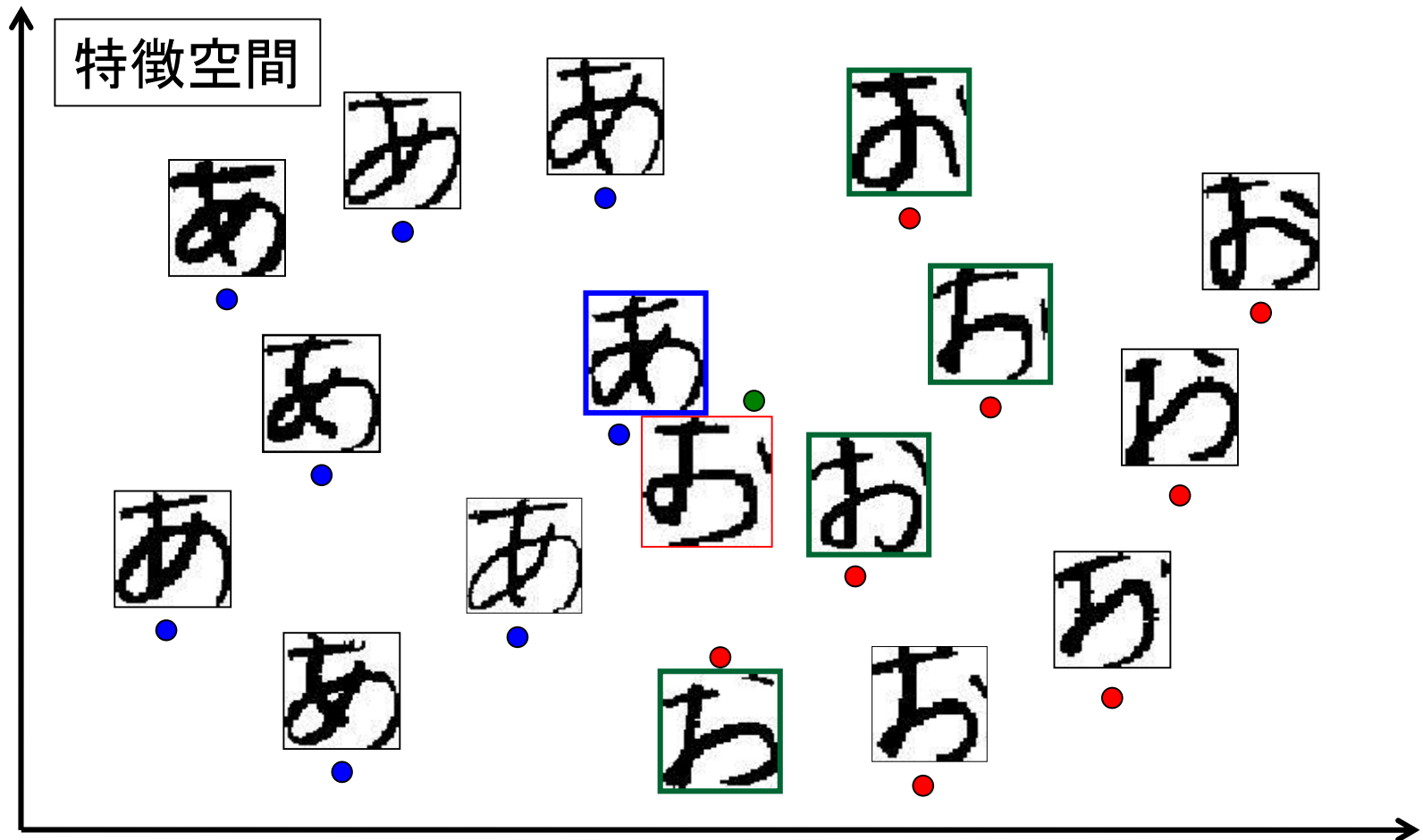
四番目まで近いプロトタイプを調べると「あ」が1個,「お」が3個
→「お」と認識



k近傍法⑪

k=5 の場合

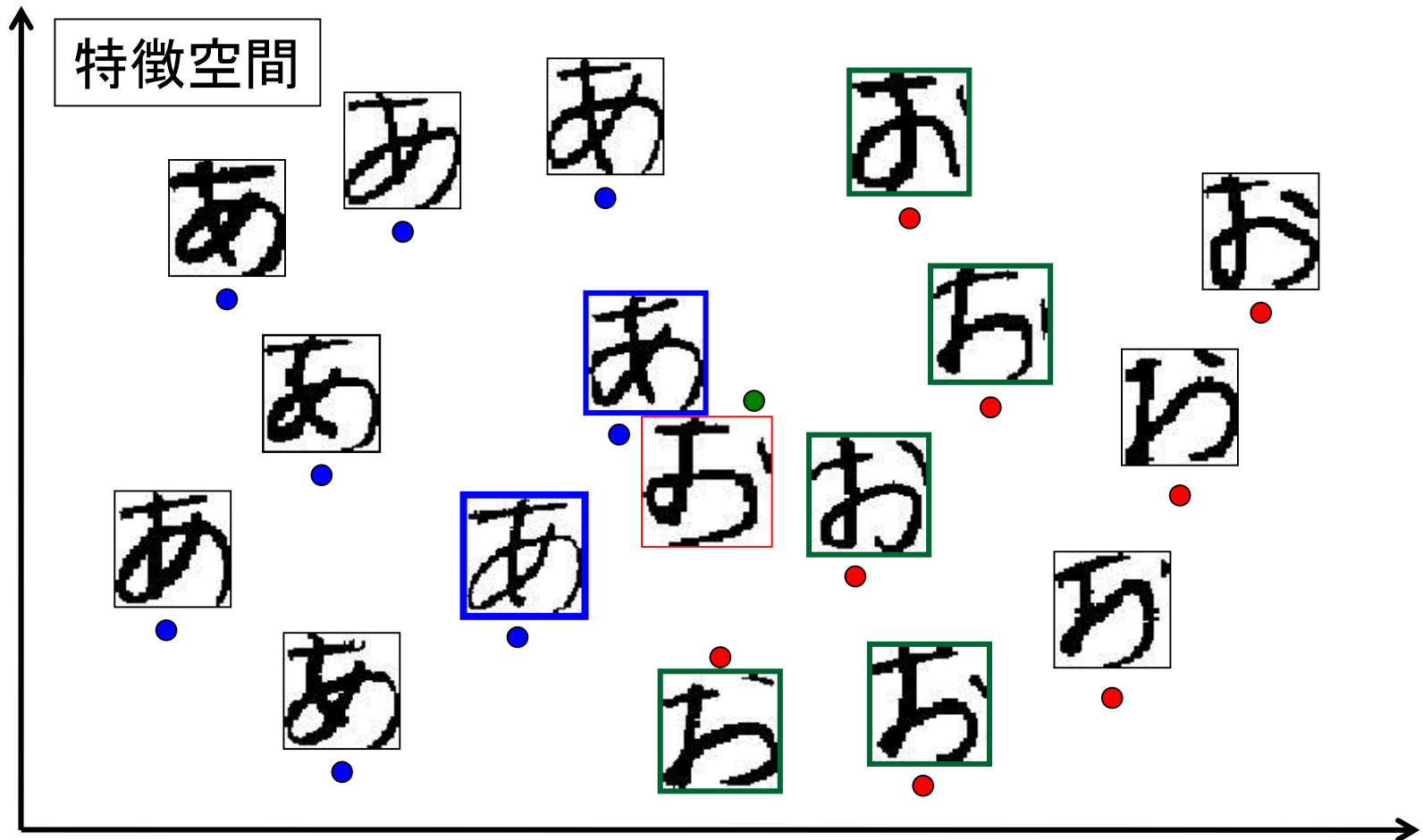
五番目まで近いプロトタイプを調べると「あ」が1個,「お」が4個
→「お」と認識



k近傍法⑫

k=7 の場合

七番目まで近いプロトタイプを調べると「あ」が2個,「お」が5個
→「お」と認識



k 近傍法のまとめ

- 一つの文字画像について複数個のプロトタイプを用意
- k 番目までに近いプロトタイプを調べる
- k 個の候補の多数決によって最終的な認識結果を決定
- $k=1$ の場合は、最近傍法と同等
- 問題点
 - プロトタイプ数の増加によって計算量が増加

実習問題(表計算)

k近傍法

10-18...

ホーム 挿入 ページレイアウト データ ツール 表示

貼り付け フォント 配置 数値 スタイル セル

クラス1のプロトタイプ

クラス1		
	x1	x2
1-1	9.1	5.4
1-2	10.4	6.8
1-3	8.2	5.3
1-4	7.5	4.7
1-5	9.7	5.2
1-6	5.9	4.5
1-7	6.5	3.2
1-8	4.5	7.2
1-9	8.2	3.8
1-10	7.4	7.1

クラス2のプロトタイプ

クラス2		
	x1	x2
2-1	2.3	3.1
2-2	0.7	1.4
2-3	2.5	3.3
2-4	1.1	3.3
2-5	2.9	6.1
2-6	1	1.2
2-7	4.2	2
2-8	3.8	5
2-9	5	1.3
2-10	3	6

未知データ1~3

未知データ1		
	x1	x2
		5
		4

未知データ2		
	x1	x2
		6
		6

未知データ3		
	x1	x2
		4
		4

コマンド 100%

■ 実習(k近傍法)

- ❑ クラス1 10個のプロトタイプ
- ❑ クラス2 10個のプロトタイプ
- ❑ 特徴ベクトル(x_1, x_2)
- ❑ 未知データ1~3がどのクラスに属するかを調べる
- ❑ 類似度はユークリッド距離

10-18.xlsx - Microsoft Excel

ホーム 挿入 ページレイアウト 数式 データ 校閲 表示

貼り付け グリッドボード フォント 配置 数値 スタイル セル 編集

TRANSPOSE X ✓ f =SQRT((C29-B3)^2+(\$C\$30-C3)^2)

	A	B	C	D	E	F	G
1		クラス1			未知データ1との距離		
2		x1	x2		プロトタイプ距離		
3	1-1	9.1	5.4		1-1	=SQRT((C29-B3)^2+(\$C\$30-C3)^2)	
4	1-2	10.4	6.8		1-2		
5	1-3	8.2	5.3		1-3		
6	1-4	7.5	4.7		1-4		
7	1-5	9.7	5.2		1-5		
8	1-6	5.9	4.5		1-6		
9	1-7	6.5	3.2		1-7		
10	1-8	4.5	7.2		1-8		
11	1-9	8.2	3.8		1-9		
12	1-10	7.4	7.1		1-10		
13					2-1		
14		クラス2			2-2		
15		x1	x2		2-3		
16	2-1	2.3	3.1		2-4		
17	2-2	0.7	1.4		2-5		
18	2-3	2.5	3.3		2-6		
19	2-4	1.1	3.3		2-7		
20	2-5	2.9	6.1		2-8		
21	2-6	1	1.2		2-9		
22	2-7	4.2	2		2-10		
23	2-8	3.8	5				
24	2-9	5	1.3				
25	2-10	3	6				
26							
27							
28		未知データ1					
29		x1	5				
30		x2	4				
31							
32		未知データ2					
33		x1	6				
34		x2	6				
35							
36		未知データ3					
37		x1	4				
38		x2	4				

編集 近傍法 マハラノビス Sheet3 100%

ユークリッド距離の計算

クラス1の各プロトタイプと未知データ1とのユークリッド距離を計算

セルF3

=SQRT((\$C\$29-B3)^2+(\$C\$30-C3)^2)

絶対参照

セルF3 → 右クリック → 「コピー」

F4～F12 を選択 → 右クリック → 「貼り付け」

10-18.xlsx - Microsoft Excel

ホーム 挿入 ページレイアウト 数式 データ 校閲 表示

貼り付け 配置 数値 スタイル セル 編集

TRANSPOSE =SQRT((\$C\$29-B16)^2+(\$C\$30-C16)^2)

	A	B	C	E	F	G
2		x1	x2	プロトタイプ	距離	
3	1-1	9.1	5.4	1-1	4.332436	
4	1-2	10.4	6.8	1-2	6.082763	
5	1-3	8.2	5.3	1-3	3.453983	
6	1-4	7.5	4.7	1-4	2.596151	
7	1-5	9.7	5.2	1-5	4.850773	
8	1-6	5.9	4.5	1-6	1.029563	
9	1-7	6.5	3.2	1-7	1.7	
10	1-8	4.5	7.2	1-8	3.238827	
11	1-9	8.2	3.8	1-9	3.206244	
12	1-10	7.4	7.1	1-10	3.920459	
13				2-1	=SQRT((\$C\$29-B16)^2+(\$C\$30-C16)^2)	
14		クラス2		2-2		
15		x1	x2	2-3		
16	2-1	2.3	3.1	2-4		
17	2-2	0.7	1.4	2-5		
18	2-3	2.5	3.3	2-6		
19	2-4	1.1	3.3	2-7		
20	2-5	2.9	6.1	2-8		
21	2-6	1	1.2	2-9		
22	2-7	4.2	2	2-10		
23	2-8	3.8	5			
24	2-9	5	1.3			
25	2-10	3	6			
26						
27						
28		未知データ1				
29		x1	5			
30		x2	4			
31						
32		未知データ2				
33		x1	6			
34		x2	6			
35						
36		未知データ3				
37		x1	4			
38		x2	4			
39						

近傍法 マハラビス Sheet3

ユークリッド距離の計算

クラス2の各プロトタイプと未知データ1とのユークリッド距離を計算

セルF13

=SQRT((\$C\$29-B16)^2+(\$C\$30-C16)^2)

絶対参照

セルF13 → 右クリック → 「コピー」
F14～F22 を選択 → 右クリック → 「貼り付け」

10-18.xlsx - Microsoft Excel

ホーム 挿入 ページ レイアウト 数式 データ 校閲 表示

MS Pゴシック 11

貼り付け クリップボード

配置 数値 スタイル セル 編集

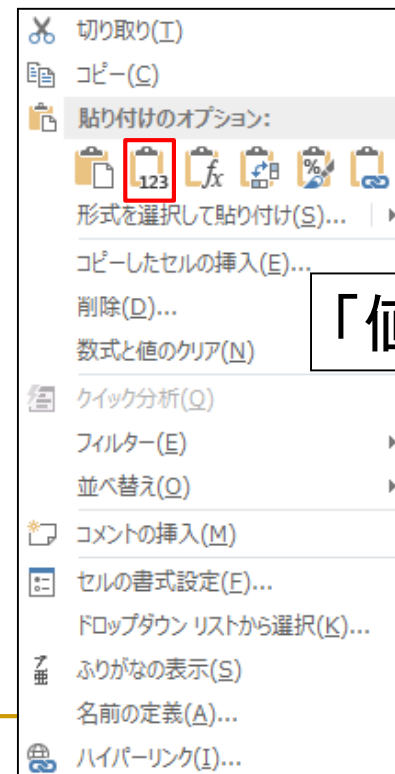
	A	B	C	D	E	F
2		x1	x2		プロトタイプ距離	
3	1-1	9.1	5.4		1-1	4.332436
4	1-2	10.4	6.8		1-2	6.082763
5	1-3	8.2	5.3		1-3	3.453983
6	1-4	7.5	4.7		1-4	2.596151
7	1-5	9.7	5.2		1-5	4.850773
8	1-6	5.9	4.5		1-6	1.029563
9	1-7	6.5	3.2		1-7	1.7
10	1-8	4.5	7.2		1-8	3.238827
11	1-9	8.2	3.8		1-9	3.206244
12	1-10	7.4	7.1		1-10	3.920459
13					2-1	2.84605
14		クラス2			2-2	5.024938
15		x1	x2		2-3	2.596151
16	2-1	2.3	3.1		2-4	3.962323
17	2-2	0.7	1.4		2-5	2.969848
18	2-3	2.5	3.3		2-6	4.882622
19	2-4	1.1	3.3		2-7	2.154066
20	2-5	2.9	6.1		2-8	1.56205
21	2-6	1	1.2		2-9	2.7
22	2-7	4.2	2		2-10	2.828427
23	2-8	3.8	5			
24	2-9	5	1.3			
25	2-10	3	6			
26						
27						
28		未知データ1				
29		x1	5			
30		x2	4			
31						
32		未知データ2				
33		x1	6			
34		x2	6			
35						
36		未知データ3				
37		x1	4			
38		x2	4			
39						

近傍法 マハラノビス Sheet3

平均: 3.296883683 データの個数: 40 合計: 65.93767366 100%

セル E3～F22 を選択
→「右クリック」→「コピー」

セルF25 → 右クリック
→「貼り付けのオプション」



「値(V)」を選択

10-18.xlsx - Microsoft Excel

	A	B	C	D	E	F	G
8	1-6	5.9	4.5		1-6	1.029563	
9	1-7	6.5	3.2		1-7	1.7	
10	1-8	4.5	7.2		1-8	3.238827	
11	1-9	8.2	3.8		1-9	3.206244	
12	1-10	7.4	7.1		1-10	3.920459	
13					2-1	2.84605	
14		クラス2			2-2	5.024938	
15		x1	x2		2-3	2.596151	
16	2-1	2.3	3.1		2-4	3.962323	
17	2-2	0.7	1.4		2-5	2.969848	
18	2-3	2.5	3.3		2-6	4.882622	
19	2-4	1.1	3.3		2-7	2.154066	
20	2-5	2.9	6.1		2-8	1.56205	
21	2-6	1	1.2		2-9	2.7	
22	2-7	4.2	2		2-10	2.828427	
23	2-8	3.8	5				
24	2-9	5	1.3				
25	2-10	3	6		1-1	4.33243	
26					1-2	6.08276	
27					1-3	3.45398	
28	未知データ1				1-4	2.59615	
29	x1		5		1-5	4.85077	
30	x2		4		1-6	1.02956	
31					1-7	1	
32	未知データ2				1-8	3.23882	
33	x1		6		1-9	3.20624	
34	x2		6		1-10	3.92045	
35					2-1	2.84605	
36	未知データ3				2-2	5.02493	
37	x1		4		2-3	2.59615	
38	x2		4		2-4	3.96232	
39					2-5	2.96984	
40					2-6	4.88262	
41					2-7	2.15406	
42					2-8	1.56205	
43					2-9	2.7	
44					2-10	2.82842	
45							

並べ替え

レベルの追加(A) レベルの削除(D) レベルのコピー(C) オプション(O)... ☐ 先頭行をデータの見出しとして使用する(H)

列 並べ替えのキー 順序

最優先されるキー 列 F 値 昇順

OK キャンセル

距離のソーティング

セル E25～F44 を選択

ツールボタン「ホーム」
→「並び替えとフィルター」
→「ユーザ設定の並び替え」

列F

昇順

結果

k=7の場合
「1-6」「2-8」「1-7」「2-7」「1-4」「2-3」「2-9」
→ クラス2

未知データ2について調べて下さい

10-18.xlsx - Microsoft Excel

ホーム 挿入 ページレイアウト 数式 データ 校閲 表示

標準 数値 セル 編集

TRANSPOSE X ✓ $=\text{SQRT}((C33-B3)^2+(C34-C3)^2)$

	A	B	C	D	E	F	G	H	I
1		クラス1			未知データ1との距離			未知データ2との距離	
2		x1	x2		プロトタイプ距離			プロトタイプ距離	
3	1-1	9.1	5.4		1-1	4.332436		1-1	$=\text{SQRT}((C33-B3)^2+(C34-C3)^2)$
4	1-2	10.4	6.8		1-2	6.082763		1-2	
5	1-3	8.2	5.3		1-3	3.453983		1-3	
6	1-4	7.5	4.7		1-4	2.596151		1-4	
7	1-5	9.7	5.2		1-5	4.850773		1-5	
8	1-6	5.9	4.5		1-6	1.029563		1-6	
9	1-7	6.5	3.2		1-7	1.7		1-7	
10	1-8	4.5	7.2		1-8	3.238827		1-8	
11	1-9	8.2	3.8		1-9	3.206244		1-9	
12	1-10	7.4	7.1		1-10	3.920459		1-10	
13					2-1	2.84605		2-1	
14		クラス2			2-2	5.024938		2-2	
15		x1	x2		2-3	2.596151		2-3	
16	2-1	2.3	3.1		2-4	3.962323		2-4	
17	2-2	0.7	1.4		2-5	2.969848		2-5	
18	2-3	2.5	3.3		2-6	4.882622		2-6	
19	2-4	1.1	3.3		2-7	2.154066		2-7	
20	2-5	2.9	6.1		2-8	1.56205		2-8	
21	2-6	1	1.2		2-9	2.7		2-9	
22	2-7	4.2	2		2-10	2.828427		2-10	
23	2-8	3.8	5						
24	2-9	5	1.3						
25	2-10	3	6		1-6	1.029563			
26					2-8	1.56205			
27					1-7	1.7			
28		未知データ1			2-7	2.154066			
29		x1	5		1-4	2.596151			
30		x2	4		2-3	2.596151			
31					2-9	2.7			
32		未知データ2			2-10	2.828427			
33		x1	6		2-1	2.84605			
34		x2	6		2-5	2.969848			
35					1-9	3.206244			
36		未知データ3			1-8	3.238827			
37		x1	4		1-3	3.453983			
38		x2	4		1-10	3.920459			

編集 10-18.xlsx マハラビス Sheet3 100%

クラス1の各プロトタイプと未知データ2とのユークリッド距離を計算

セルI3

$=\text{SQRT}((\$C\$33-B3)^2+(\$C\$34-C3)^2)$

I4~I12にコピー

クラス2の各プロトタイプと未知データ2とのユークリッド距離を計算

セルI13

$=\text{SQRT}((\$C\$33-B16)^2+(\$C\$34-C16)^2)$

I14~I22にコピー

10-18.xlsx - Microsoft Excel

ホーム 挿入 ページ レイアウト 数式 データ 校閲 表示

MS Pゴシック 11

貼り付け クリップボード

数値 スタイル セル

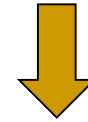
配置 編集

G25

	C	D	E	F	G	H	I
9	3.2		1-7	1.7		1-7	2.844293
10	7.2		1-8	3.238827		1-8	1.920937
11	3.8		1-9	3.206244		1-9	3.11127
12	7.1		1-10	3.920459		1-10	1.780449
13			2-1	2.84605		2-1	4.701064
14			2-2	5.024938		2-2	7.017834
15	x2		2-3	2.596151		2-3	4.420407
16	3.1		2-4	3.962323		2-4	5.59464
17	1.4		2-5	2.969848		2-5	3.101612
18	3.3		2-6	4.882622		2-6	6.931089
19	3.3		2-7	2.154066		2-7	4.386342
20	6.1						
21	1.2						
22	2						
23	5						
24	1.3						
25	6		1-6	1.029563		1-6	1.50333
26			2-8	1.56205		1-10	1.780449
27			1-7	1.7		1-8	1.920937
28	1		2-7	2.154066		1-4	1.984943
29	5		1-4	2.596151		1-3	2.308679
30	4		2-3	2.596151		2-8	2.416609
31			2-9	2.7		1-7	2.844293
32	2		2-10	2.828427		2-10	3
33	6		2-1	2.84605		2-5	3.101612
34	6		2-5	2.969848		1-9	3.11127
35			1-9	3.206244		1-1	3.157531
36	3		1-8	3.238827		1-5	3.785499
37	4		1-3	3.453983		2-7	4.386342
38	4		1-10	3.920459		2-3	4.420407
39			2-4	3.962323		1-2	4.472136
40			1-1	4.332436		2-1	4.701064
41			1-5	4.850773		2-9	4.805206
42			2-6	4.882622		2-4	5.59464
43			2-2	5.024938		2-6	6.931089
44			1-2	6.082763		2-2	7.017834
45							
46							

未知データ2の結果

H3~I22の値をH25~I44にコピー
I列でソーティング



k=1の場合
「1-6」→ クラス1

k=3の場合
「1-6」「1-10」「1-8」→ クラス1

k=5の場合
「1-6」「1-10」「1-8」「1-4」「1-3」
→ クラス1

未知データ3について調べて下さい

クラス1の各プロトタイプと未知データ3とのユークリッド距離を計算

セルL3

$=\text{SQRT}((\$C\$37-B3)^2+(\$C\$38-C3)^2)$

L4～L12にコピー

クラス2の各プロトタイプと未知データ3とのユークリッド距離を計算

セルIL13

$=\text{SQRT}((\$C\$37-B16)^2+(\$C\$38-C16)^2)$

L14～L22にコピー

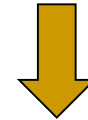
	F	G	H	I	J	K	L
1	1との距離		未知データ2との距離			未知データ3との距離	
2	距離		プロトタイプ距離			プロトタイプ距離	
3	4.332436		1-1	3.157531		1-1	$=\text{SQRT}((\$C\$37-B3)^2+(\$C\$38-C3)^2)$
4	6.082763		1-2	4.472136		1-2	
5	3.453983		1-3	2.308679		1-3	
6	2.596151		1-4	1.984943		1-4	
7	4.850773		1-5	3.785499		1-5	
8	1.029563		1-6	1.50333		1-6	
9	1.7		1-7	2.844293		1-7	
10	3.238827		1-8	1.920937		1-8	
11	3.206244		1-9	3.11127		1-9	
12	3.920459		1-10	1.780449		1-10	
13	2.84605		2-1	4.701064		2-1	
14	5.024938		2-2	7.017834		2-2	
15	2.596151		2-3	4.420407		2-3	
16	3.962323		2-4	5.59464		2-4	
17	2.969848		2-5	3.101612		2-5	
18	4.882622		2-6	6.931089		2-6	
19	2.154066		2-7	4.386342		2-7	
20	1.56205		2-8	2.416609		2-8	
21	2.7		2-9	4.805206		2-9	
22	2.828427		2-10	3		2-10	
23							
24							
25	1.029563		1-6	1.50333			
26	1.56205		1-10	1.780449			
27	1.7		1-8	1.920937			
28	2.154066		1-4	1.984943			
29	2.596151		1-3	2.308679			
30	2.596151		2-8	2.416609			
31	2.7		1-7	2.844293			
32	2.828427		2-10	3			
33	2.84605		2-5	3.101612			
34	2.969848		1-9	3.11127			
35	3.206244		1-1	3.157531			
36	3.238827		1-5	3.785499			
37	3.453983		2-7	4.386342			
38	3.920459		2-3	4.420407			

10-18.xlsx - Microsoft Excel

	F	G	H	I	J	K	L
8	1.029563		1-6	1.50333		1-6	5.921149
9	1.7		1-7	2.844293		1-7	6.549046
10	3.238827		1-8	1.920937		1-8	5.521775
11	3.206244		1-9	3.11127		1-9	8.202439
12	3.920459		1-10	1.780449		1-10	8.023092
13	2.84605		2-1	4.701064		2-1	1.923538
14	5.024938		2-2	7.017834		2-2	4.20119
15	2.596151		2-3	4.420407		2-3	1.655295
16	3.962323		2-4	5.59464		2-4	2.983287
17	2.969848		2-5	3.101612		2-5	2.370654
18	4.882622		2-6	6.931089		2-6	4.103657
19	2.154066		2-7	4.386342		2-7	2.009975
20	1.56205		2-8	2.416609		2-8	1.019804
21	2.7		1-10	1.780449		2-3	1.655295
22	2.828427		1-8	1.920937		2-1	1.923538
23			1-4	1.984943		2-7	2.009975
24			1-3	2.308679		2-10	2.236068
25	1.029563		2-8	2.416609		2-5	2.370654
26	1.56205		1-7	2.844293		2-9	2.879236
27	1.7		2-10	3		2-4	2.983287
28	2.154066		2-5	3.101612		2-6	4.103657
29	2.596151		1-9	3.11127		2-2	4.20119
30	2.596151		1-1	3.157531		1-1	5.288667
31	2.7		1-5	3.785499		1-8	5.521775
32	2.828427		2-7	4.386342		1-6	5.921149
33	2.84605		2-3	4.420407		1-7	6.549046
34	2.969848		1-2	4.472136		1-2	6.9857
35	3.206244		2-1	4.701064		1-4	7.532596
36	3.238827		2-9	4.805206		1-10	8.023092
37	3.453983		2-4	5.59464		1-9	8.202439
38	3.920459		2-6	6.931089		1-3	8.302409
39	3.962323		2-2	7.017834		1-5	9.773945
40	4.332436						
41	4.850773						
42	4.882622						
43	5.024938						
44	6.082763						
45							

未知データ3の結果

K3~L22の値をK25~L44にコピー
1列でソーティング

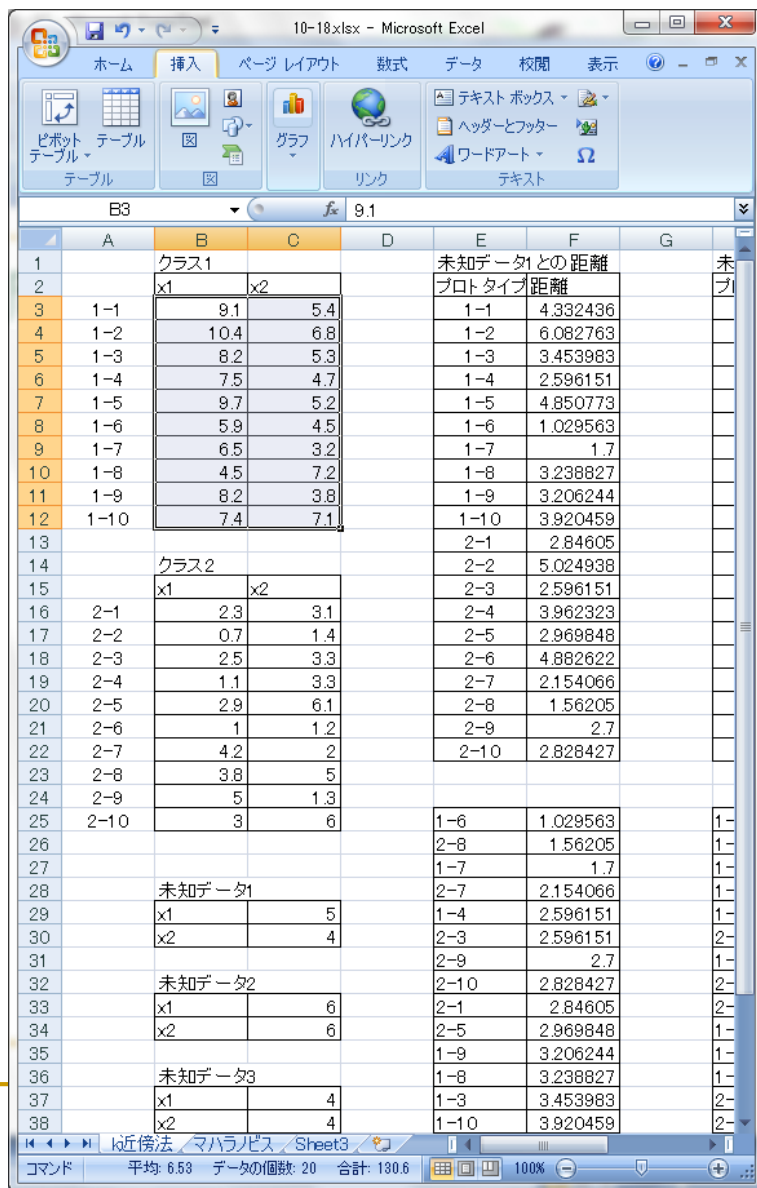


k=1の場合
「2-8」→ クラス2

k=3の場合
「2-8」「2-3」「2-1」→ クラス2

k=5の場合
「2-7」「2-3」「2-1」「2-10」「2-5」
→ クラス2

散布図の作成①

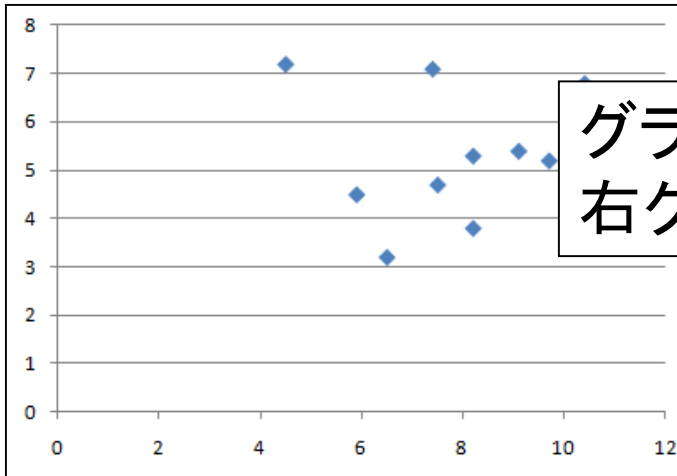


	A	B	C	D	E	F	G
1		クラス1			未知データ1との距離		
2		x1	x2		プロトタイプ距離		
3	1-1	9.1	5.4		1-1	4.332436	
4	1-2	10.4	6.8		1-2	6.082763	
5	1-3	8.2	5.3		1-3	3.453983	
6	1-4	7.5	4.7		1-4	2.596151	
7	1-5	9.7	5.2		1-5	4.850773	
8	1-6	5.9	4.5		1-6	1.029563	
9	1-7	6.5	3.2		1-7	1.7	
10	1-8	4.5	7.2		1-8	3.238827	
11	1-9	8.2	3.8		1-9	3.206244	
12	1-10	7.4	7.1		1-10	3.920459	
13					2-1	2.84605	
14		クラス2			2-2	5.024938	
15		x1	x2		2-3	2.596151	
16	2-1	2.3	3.1		2-4	3.962323	
17	2-2	0.7	1.4		2-5	2.969848	
18	2-3	2.5	3.3		2-6	4.882622	
19	2-4	1.1	3.3		2-7	2.154066	
20	2-5	2.9	6.1		2-8	1.56205	
21	2-6	1	1.2		2-9	2.7	
22	2-7	4.2	2		2-10	2.828427	
23	2-8	3.8	5				
24	2-9	5	1.3				
25	2-10	3	6		1-6	1.029563	1-
26					2-8	1.56205	1-
27					1-7	1.7	1-
28		未知データ1			2-7	2.154066	1-
29		x1	5		1-4	2.596151	1-
30		x2	4		2-3	2.596151	2-
31					2-9	2.7	1-
32		未知データ2			2-10	2.828427	2-
33		x1	6		2-1	2.84605	2-
34		x2	6		2-5	2.969848	1-
35					1-9	3.206244	1-
36		未知データ3			1-8	3.238827	1-
37		x1	4		1-3	3.453983	2-
38		x2	4		1-10	3.920459	2-

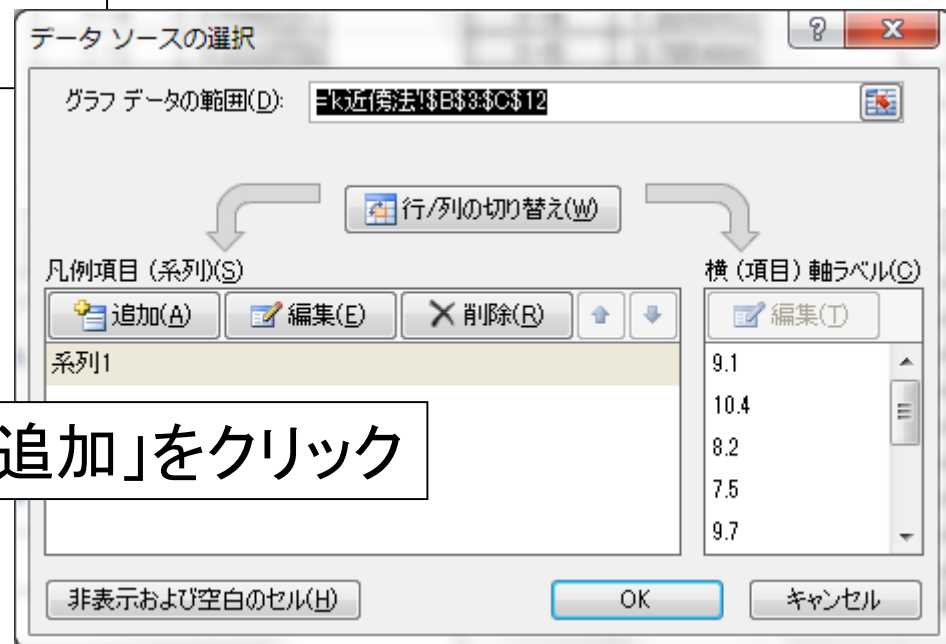
セルB3～C12 を選択
→ ツールバー「挿入」
→ 「散布図」



散布図の作成②



グラフ上で、
右クリック→「データの選択」



散布図の作成③

系列の編集

系列名(N):

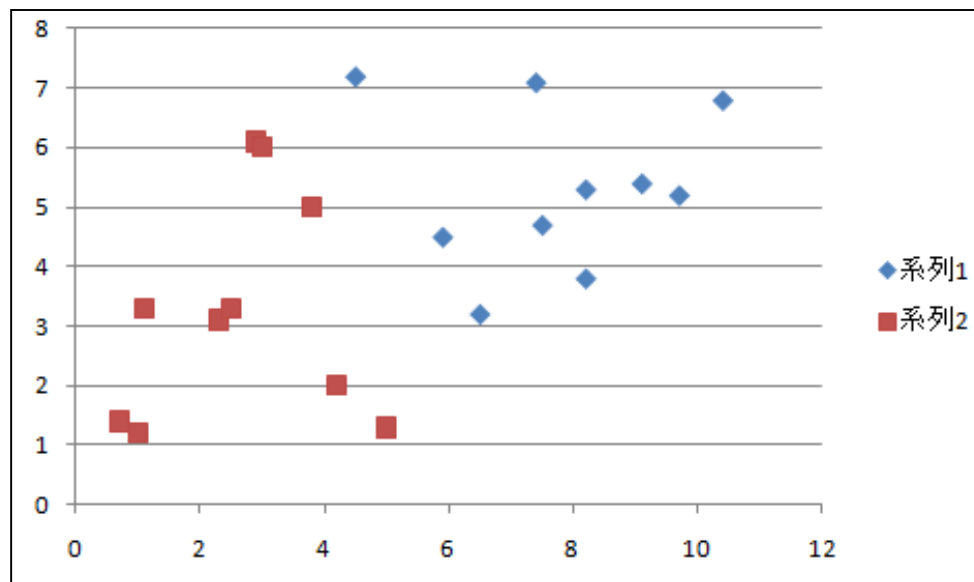
系列 X の値(X):

系列 Y の値(Y):

OK キャンセル

クリック → 「B16～B25」を選択

クリック → 「C16～C25」を選択



散布図の作成④

■ 「追加」をクリックし，未知データを入力

未知データ1

系列の編集

系列名(N): データ範囲の選択

系列 X の値(X): = 5

系列 Y の値(Y): = 1

OK キャンセル

Xの値: C29
Yの値: C30

未知データ2

系列の編集

系列名(N): データ範囲の選択

系列 X の値(X): = 6

系列 Y の値(Y): = 6

OK キャンセル

Xの値: C33
Yの値: C34

未知データ3

系列の編集

系列名(N): データ範囲の選択

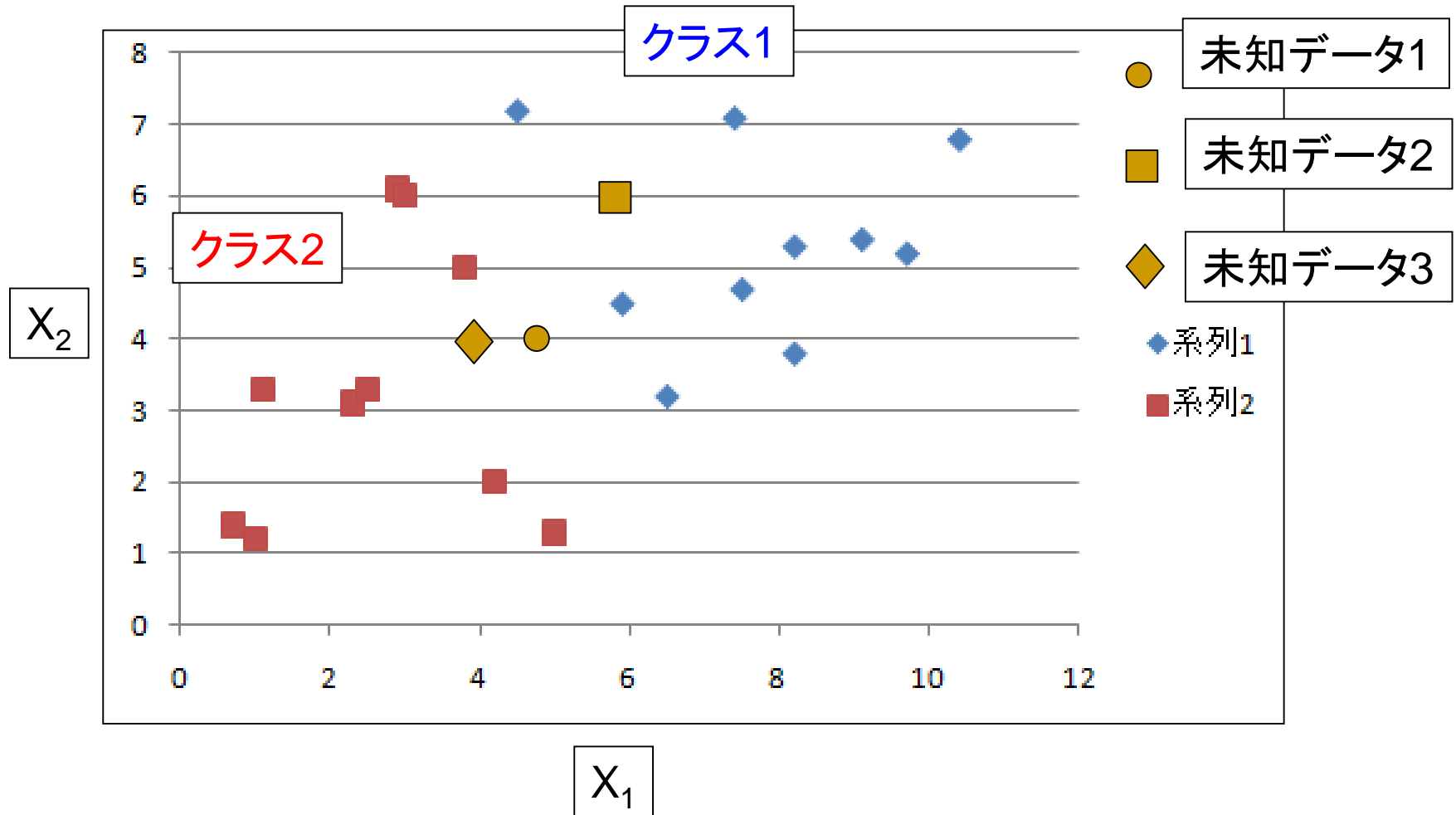
系列 X の値(X): = 4

系列 Y の値(Y): = 4

OK キャンセル

Xの値: C37
Yの値: C38

二つのクラスの分布



k 近傍法のプログラム

数字認識 (digitsデータベース)

k近傍法

■ digitsデータベース

用途	クラス分類
データ数	1797
特徴量	画素数:64(8×8) 値:0~16
目的変数	10

数字	データ数
0	178
1	182
2	177
3	183
4	181
5	182
6	181
7	179
8	174
9	180

nearest_neighbor.py

```
import numpy as np
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
```

k近傍法のためにimportが必要



k近傍法

K = 5

近傍数k

データのロード

digits = datasets.load_digits()

数字画像(digits)の読み込み

特徴量 (1797, 8, 8)

image = digits.images

total, x_size, y_size = image.shape

3次元→2次元に変換

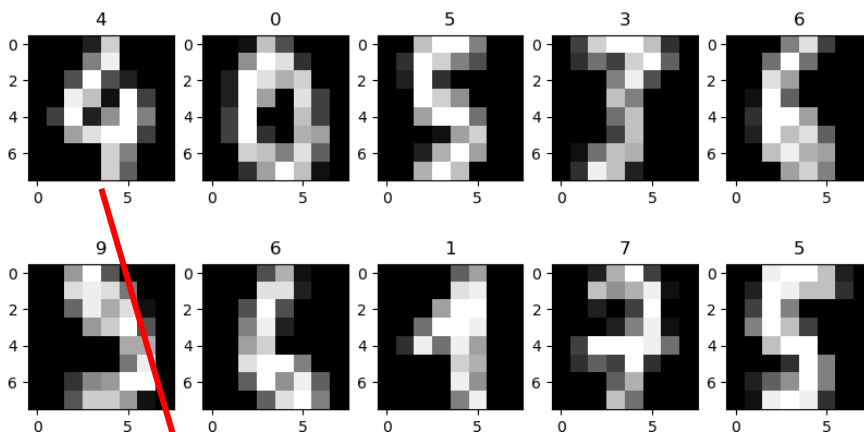
image = np.reshape(image, [total, x_size*y_size])

```
total, x_size, y_size = image.shape
```

配列の大きさ(3次元)
(total, z_size, y_size)

横: x_size

縦: y_size



個数: total

```
image = np.reshape( image , [total,x_size*y_size] )
```

total

x_size*y_size

配列の大きさ(2次元)
(total, z_size*y_size)

目的変数

```
label = digits.target
```

学習データ, テストデータ

```
train_data, test_data, train_label, test_label = train_test_split(image, label,  
test_size=0.5, random_state=None)
```

ホールドアウト法

```
model = KNeighborsClassifier(n_neighbors=K, algorithm='brute',  
metric='minkowski', p=2)
```

n_neighbors
近傍数k

algorithm
'brute' → 全探索(デフォルト)

metric
'minkowski' → ミンコスキー距離
(デフォルトはp=2のユークリッド距離)

学習

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

p=1の場合→マンハッタン距離
p=2の場合→ユークリッド距離(デフォルト)

予測

```
predict = model.predict(test_data)
```

```
distance, result = model.kneighbors(test_data, n_neighbors=K)
```

K個の候補, 距離, 予測結果, 正解の表示

```
for i in range(len(test_data)):
    for j in range( K ):
        print( "{0:4d}({1})".format( result[i,j], train_label[ result[i,j] ] ) , end=" ")
    print( " [ " , end="" )
    for j in range( K ):
        print( "{0:5.2f}".format( distance[i,j] ) , end=" ")
    print( " ] -> {0} [ {1} ]".format( predict[i] , test_label[i] ) )

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

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

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

result[i,j]

i番目のデータにおける
第j候補のデータ番号

train_label[result[i,j]]

第j候補のラベル

distance[i,j]

i番目のデータにおける
第j候補との距離

評価指標の表示

k近傍法

n_neighbors
近傍数k

algorithm
'brute' → 全探索(デフォルト)
高速化したい場合, 'kd_tree' または 'ball_tree'

`KNeighborsClassifier(n_neighbors=K, algorithm='brute',
metric='minkowski', p=2)`

metric
'minkowski' → ミンコスキー距離
(デフォルトはp=2のユークリッド距離)

p=1の場合 → マンハッタン距離
p=2の場合 → ユークリッド距離(デフォルト)

簡単に使いたい場合 (K=4)

```
model = KNeighborsClassifier(n_neighbors=4)
```

予測

学習

`fit`(学習データ, 学習データに対する正解ラベル)

予測

`predict`(予測したいデータ)

学習

`model.fit`(train_data, train_label)

予測(テストデータの場合)

`predict = model.predict`(test_data)

予測(学習データの場合)

`predict = model.predict`(train_data)

予測

```
distance , result = model.kneighbors(test_data,n_neighbors=K)
```

distance
近傍数Kまでの距離

result
近傍数Kまでのデータ番号

n_neighbors
近傍数K

K=5

第1候補

第2候補

第3候補

第4候補

第5候補

distance

561(4)	467(4)	393(4)	505(4)	697(4)	[13.30 23.56 23.90 23.98 24.82]
813(5)	171(5)	582(5)	770(5)	546(5)	[26.40 26.94 29.46 30.18 30.51]
751(2)	709(2)	506(2)	45(2)	109(2)	[10.10 15.23 16.40 24.84 24.98]
852(8)	254(8)	161(2)	734(6)	407(6)	[22.61 27.89 28.81 30.77 30.81]
259(0)	82(0)	214(0)	463(0)	179(0)	[19.54 19.75 20.02 20.22 20.52]
854(8)	705(8)	889(8)	28(8)	584(8)	[13.78 17.75 18.63 21.75 23.30]
477(0)	706(0)	338(0)	381(0)	172(0)	[13.64 14.18 14.49 15.75 15.91]
193(0)	100(0)	735(0)	377(0)	335(0)	[16.49 18.06 18.28 18.36 18.87]
88(4)	718(4)	208(4)	846(4)	309(4)	[19.52 19.60 20.00 20.62 22.85]
732(0)	311(0)	604(0)	259(0)	179(0)	[12.21 14.49 14.93 15.13 15.23]
374(9)	170(9)	826(9)	715(9)	744(9)	[14.42 16.67 17.69 19.54 21.21]

result

ラベル

第1候補

第5候補

評価指標の計算

accuracy

```
from sklearn.metrics import accuracy_score  
accuracy_score(正解ラベル, 予測結果)
```

accuracy, precision, recall, F値

```
from sklearn.metrics import classification_report  
classification_report(正解ラベル, 予測結果)
```

混同行列

```
from sklearn.metrics import confusion_matrix  
confusion_matrix(正解ラベル, 予測結果)
```

実行結果①(K=5の場合)

test_label
(正解ラベル)

result	ラベル	distance	predict (予測結果)	
C:\Windows\system32\cmd.exe				
C:\home\chino\ML-2019\最近傍法>python nearest_neighbor.py				
531(7)	374(7)	891(7)	654(7)	293(7) [15.84 19.39 20.32 21.73 21.93] -> 7 [7]
569(9)	494(9)	652(9)	161(9)	823(9) [19.85 21.59 22.18 22.20 22.20] -> 9 [9]
710(1)	219(1)	128(1)	301(1)	804(1) [17.61 17.69 19.31 19.44 20.95] -> 1 [1]
488(0)	737(0)	27(0)	81(0)	261(0) [20.64 21.70 21.73 21.98 22.67] -> 0 [0]
250(6)	157(6)	9(6)	665(6)	609(6) [18.41 18.73 19.39 19.57 19.60] -> 6 [6]
531(7)	308(7)	585(7)	293(7)	374(7) [15.26 22.91 23.00 23.39 25.22] -> 7 [7]
412(6)	699(6)	57(6)	799(6)	671(6) [14.39 15.62 17.46 18.55 18.79] -> 6 [6]
805(3)	107(3)	76(3)	622(3)	760(3) [23.87 23.92 25.75 26.44 27.22] -> 3 [3]
808(9)	494(9)	558(9)	170(9)	390(9) [17.66 19.03 20.71 22.00 24.27] -> 9 [9]
762(7)	240(7)	467(7)	504(7)	644(7) [14.21 14.56 15.07 17.09 17.15] -> 7 [7]
577(0)	261(0)	516(0)	864(0)	304(0) [15.33 16.25 17.69 18.17 18.52] -> 0 [0]
220(6)	873(6)	609(6)	827(6)	860(6) [12.45 13.49 13.89 14.32 15.62] -> 6 [6]
207(9)	246(9)	652(9)	161(9)	596(0) [25.53 26.96 27.07 28.32 28.57] -> 9 [9]
821(0)	864(0)	419(0)	737(0)	98(0) [13.75 14.28 15.68 17.00 17.64] -> 0 [0]
270(0)	159(0)	362(0)	216(0)	206(0) [12.77 18.14 18.33 18.68 19.13] -> 0 [0]
318(0)	276(0)	424(0)	584(0)	451(0) [15.91 16.40 17.35 17.46 17.61] -> 0 [0]
517(1)	774(1)	179(1)	560(1)	693(1) [20.15 20.62 20.86 23.13 23.49] -> 1 [1]
884(8)	245(8)	213(8)	307(8)	441(8) [25.59 26.44 26.61 27.28 27.31] -> 8 [8]
84(7)	447(7)	582(7)	266(7)	467(7) [19.16 20.54 20.81 21.28 21.31] -> 7 [7]
697(3)	110(3)	383(3)	837(3)	503(3) [18.57 19.62 20.86 21.59 22.56] -> 3 [3]
817(9)	719(9)	498(9)	343(9)	65(9) [17.52 22.36 24.76 25.30 27.82] -> 9 [9]
743(7)	123(7)	499(7)	581(7)	844(7) [24.10 31.78 35.21 35.99 37.03] -> 7 [7]
100(2)	166(2)	13(2)	435(2)	707(2) [13.78 21.95 22.09 22.11 23.32] -> 2 [2]
275(5)	688(5)	724(5)	861(5)	624(5) [17.09 21.95 26.85 27.00 27.53] -> 5 [5]
220(6)	827(6)	331(6)	873(6)	860(6) [11.45 11.96 13.23 13.78 15.36] -> 6 [6]
601(2)	389(2)	692(2)	751(2)	809(2) [13.53 17.12 20.17 21.98 22.07] -> 2 [2]
662(5)	217(5)	793(5)	373(5)	152(5) [20.47 20.86 25.98 26.53 27.18] -> 5 [5]

実行結果②

cmd.exe C:\Windows\system32\cmd.exe

[予測結果]

	precision	recall	f1-score	support
0	1.00	1.00	1.00	80
1	0.95	1.00	0.98	100
2	1.00	0.96	0.98	94
3	0.97	1.00	0.98	91
4	0.99	0.98	0.98	89
5	1.00	0.97	0.98	91
6	1.00	1.00	1.00	83
7	0.98	0.99	0.98	87
8	0.95	0.93	0.94	85
9	0.96	0.97	0.96	99
accuracy			0.98	899
macro avg	0.98	0.98	0.98	899
weighted avg	0.98	0.98	0.98	899

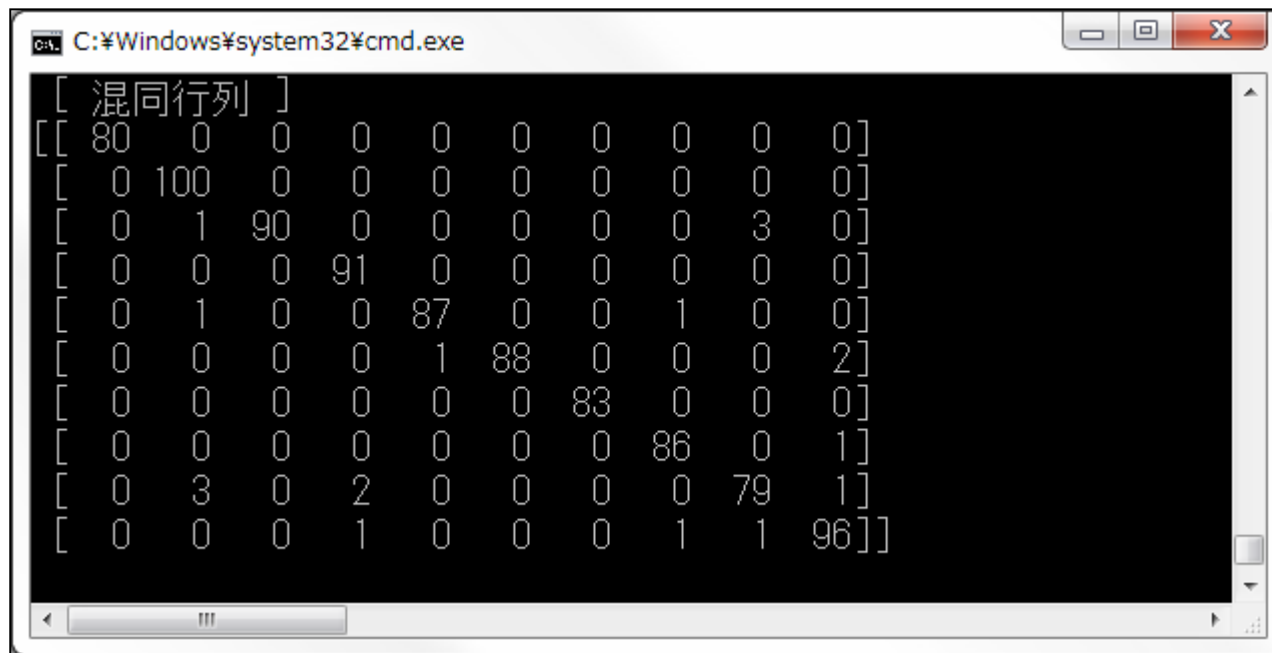
[正解率]
0.978865406006674

precision
recall
f値
accuracy

accuracy

実行結果③

混同行列



```
C:\Windows\system32\cmd.exe
[ 混同行列 ]
[ 80  0  0  0  0  0  0  0  0  0]
[  0 100  0  0  0  0  0  0  0  0]
[  0  1  90  0  0  0  0  0  3  0]
[  0  0  0  91  0  0  0  0  0  0]
[  0  1  0  0  87  0  0  1  0  0]
[  0  0  0  0  1  88  0  0  0  2]
[  0  0  0  0  0  0  83  0  0  0]
[  0  0  0  0  0  0  0  86  0  1]
[  0  3  0  2  0  0  0  0  79  1]
[  0  0  0  1  0  0  0  1  1  96]
```

- digitsデータベースについては、k近傍法が非常に精度が高い

参考文献

- 舟久保登：パターン認識，共立出版（1991）
- 石井健一郎他：わかりやすいパターン認識，オーム社（1998）
- 出口光一郎：画像認識論講義，昭晃堂（2002）
- 平井有三：はじめてのパターン認識，森北出版（2012）

参考文献

- KNeighborsClassifier
 - <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>