# 機械学習基礎実践 演習テキスト

荒木雅弘

2018 年 5 月 28 日

# 第 1 章

# パターン認識概要

## 1.1　演習の目的

プロトタイプが与えられたという前提で、最近傍決定則を Scilab でコーディングし、パターン認識の基本的な手順を理解します。

## 1.2　準備

演習問題を解く際に用いる Scilab の機能を確認します。

1. 変数に行列を代入する
2. 行列のサイズを求める（`size` 関数）
3. 行列から特定の行を抜き出す
4. 行列から特定の列を抜き出す
5. ゼロ行列を作成する（`zeros` 関数）
6. 変数にベクトルを代入する
7. ベクトルの大きさを求める（`sum`,`sqrt`,`norm` 関数）
8. 2 つのベクトルの距離を求める
9. `for` 文による繰り返し
10. 変数の値を表示する（`disp` 関数）
11. 最小値を求める（`min` 関数）
12. 最小値を与える位置を求める
13. 行列を複製する（`repmat` 関数）
14. 行列を結合する
15. ベクトルを行列に変換
16. ベクトルを文字列に変換
17. 正規表現で文字列マッチング

```
--> M = [1 2 3; 4 5 6]
 M  =
    1.    2.    3.
    4.    5.    6.

--> [r c] = size(M)
```

```
 c  =
    3.
 r  =
    2.

--> size(M,'r')
 ans  =
   2.

--> size(M,'c')
 ans  =
   3.

--> M(1,:)
 ans  =
    1.    2.    3.

--> M(:,2)
 ans  =
    2.
    5.

--> zeros(2, 3)
 ans  =
    0.    0.    0.
    0.    0.    0.

--> z = zeros()
 z  =
   0.

--> z(3) = 5
 z  =
   0.
   0.
   5.

--> v1 = [1 1]
 v1  =
    1.    1.

--> v2 = [7 9]
 v2  =
    7.    9.

--> sqrt(sum(v1.^2))
 ans  =
    1.4142136

--> norm(v1)
 ans  =
    1.4142136

--> norm(v1 - v2)
 ans  =
    10.

--> for i = 1:5
-->   disp(i^2)
```

```
--> end

    1.
    4.
    9.
    16.
    25.

--> a = [5 8 7 2 3];

--> min(a)
 ans  =
    2.

--> [m, p] = min(a)
 p  =
    4.
 m  =
    2.

--> repmat(M,[1 2])
 ans  =
   1.   2.   3.   1.   2.   3.
   4.   5.   6.   4.   5.   6.

--> repmat(M,[2 1])
 ans  =
   1.   2.   3.
   4.   5.   6.
   1.   2.   3.
   4.   5.   6.

--> f=[]
 f  =
    []

--> f = [f [1 2]']
 f  =
   1.
   2.

--> f = [f [3 4]']
 f  =
   1.   3.
   2.   4.

--> matrix([1 2 3 4 5 6], [2, 3])
 ans  =
   1.   3.   5.
   2.   4.   6.

--> strcat(string([0 1 0 1 0]))
 ans  =
 01010

--> regexp(ans, '/101/')
 ans  =
   2.
```

## 実践演習 1-1

ソースコード 1.1 の（ア），（イ）を埋め，例題 1.1 の計算過程を実行する Scilab のコードを完成させよ．

ソースコード 1.1　例題 1.1

```
clear;
P = [[0,1,1,1,0,..
      1,0,0,0,1,..
      1,0,0,0,1,..
      1,0,0,0,1,..
      0,1,1,1,0]',..
     [0,0,1,0,0,..
      0,0,1,0,0,..
      0,0,1,0,0,..
      0,0,1,0,0,..
      0,0,1,0,0]',..
     [0,1,1,1,1,..
      1,0,0,1,0,..
      0,0,1,0,0,..
      0,1,0,0,0,..
      1,1,1,1,1]',..
     [0,1,1,1,0,..
      1,0,0,0,1,..
      0,0,1,1,0,..
      1,0,0,0,1,..
      0,1,1,1,0]',..
     [0,0,1,0,0,..
      0,1,0,0,0,..
      1,0,0,1,0,..
      1,1,1,1,1,..
      0,0,0,1,0]'];

x = [0,0,0,1,0,..
     0,0,0,1,0,..
     0,0,0,1,0,..
     0,0,0,1,0,..
     0,0,0,1,0]';

dist = zeros();
for i = 1:  （ア）
    dist(i) = norm(P(  （イ）  ) - x);
end

[mindist, ans] = min(dist);
disp("Ans = "+string(ans-1))
```

## 実践演習 1-2

ソースコード 1.1 の `for` ループ処理を `repmat` 関数を用いた行列演算に置き換えよ．

## 実践演習 1-3

演習問題 1.1 の指示に従い，特徴抽出を行う機能をソースコード 1.1 に追加して，特徴抽出後の特徴ベクトルを用いて識別を行え．ただし，特徴ベクトルは，縦・横の直線数の計算のみでよい．

# 第 2 章

# 前処理

## 2.1 演習の目的

画像データを対象に、パターン認識の前処理段階で行うフィルタ処理を Scilab でコーディングし、効果を確認します。

## 2.2 Scilab への画像処理モジュールインストール

Scilab で画像を扱う際には、外部モジュールである Image Processing and Computer Vision Module を使うと便利です。以下の手順でインストールしてください。

1. Scilab を起動後、メニューの [アプリケーション] から [モジュール管理] を選択し、モジュール管理画面を表示。
2. 左側ペインから [画像処理] を選択。
3. 画像処理が展開され、[Image Processing and Computer Vision Toolbox] が表示されるので、それを選択。
4. 右側ペインの [インストール] ボタンを押してインストール開始。
5. インストールが終了したら Scilab を再起動し、起動後に”Start IPCV 1.2 for Scilab 6.0”というメッセージが表示されればインストール成功。

## 2.3 準備

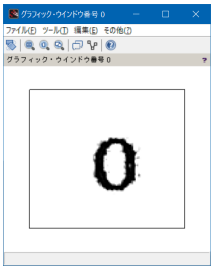
演習では、扱いやすい濃淡画像を対象データとします。画像を行列データとして読み込んだ後、各種フィルタ処理を Scilab でコーディングします。

1. 画像データを行列として読み込む（`imread` 関数）
2. 画像データを倍精度浮動小数点数に変換 （`im2double` 関数）
3. 画像を表示する（`imshow` 関数）
4. 画像をファイルに出力する（`imwrite` 関数）
5. 中間値を求める（`median` 関数）
6. 2 重ループ

```
-->im=imread('test1.pgm')  // im は整数（0〜255）の 120x120 行列
```

```
-->im2=im2double(im)    // 整数から倍精度表現（0〜1）に変換

-->imshow(im2)
```



```
-->imwrite(im2, 'out.png') // pgm, jpg, png, bmp, tiff も可

-->a = [5 8 7 2 3];

-->median(a)
 ans  =
    5.

-->b = [5 8 7 2 3 4];

-->median(b)
 ans  =
    4.5

-->for i=1:9
-->  for j=1:9
-->   printf("%3d",i*j)
-->  end
-->  printf("\n")
-->end
  1  2  3  4  5  6  7  8  9
  2  4  6  8 10 12 14 16 18
  3  6  9 12 15 18 21 24 27
  4  8 12 16 20 24 28 32 36
  5 10 15 20 25 30 35 40 45
  6 12 18 24 30 36 42 48 54
  7 14 21 28 35 42 49 56 63
  8 16 24 32 40 48 56 64 72
  9 18 27 36 45 54 63 72 81
```

## 実践演習 2-1

空欄（ア），（イ）を埋め，入力画像にメディアンフィルタを適用する Scilab のコードを完成させよ．

```
clear;

// 画像データの読み込み
im = im2double(imread('test1.pgm'));
// 2次元配列 im のサイズ取得
[h w] = size(im);
// 結果格納用の配列 resultim を用意
resultim = ones(im);
```

```
// メディアンフィルタ適用
for y = 2:h-1
    for x = 2:w-1
        resultim(   （ア）     ) = median(im(    （イ）    ));
    end
end

// 結果の表示とファイルへの出力
imshow([im, resultim])
imwrite([im, resultim], 'out.png');
```

## 実践演習 2-2

実践演習 2-1 のコードに平均値フィルタ処理を加え、原画像、メディアンフィルタ適用後、平均値フィルタ適用後の画像を並べて結果を比較せよ。

## 実践演習 2-3

入力画像に Sobel フィルタ（第 2 章講義スライド参照）を適用するコードを作成せよ。

## 実践演習 2-4

実践演習 2-3 のコードに、最大値プーリングを行う処理を追加せよ。