

第 5 章

第 4 章 前半 演習問題

5.1 演習の目的

数字画像認識を題材とした特徴抽出および最近傍決定則の Scilab による実装を行います。

5.2 データ

演習で用いる数字画像の例を図 5.1 に示します。1 枚の画像の大きさは、 120×120 です。字体は限定されていますが、撮影状態の異なる 10 種類の画像が各数字（0～9）毎に 10 枚用意されています。また、それぞれの画像には白色雑音やバイナリー雑音も加わっています。ファイル名は `number+ 正解+_+ 通し番号+.pgm`（例えば数字 0 の 5 枚目の画像は `number0_5.pgm`）の形式です。

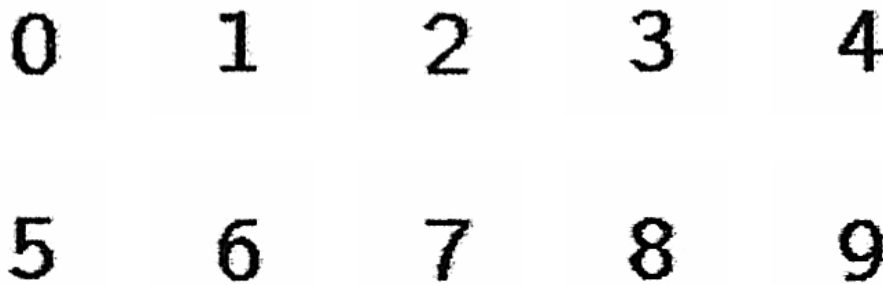


図 5.1 数字画像の例

5.3 演習の手順

5.3.1 事前処理

演習で用いる PGM 画像ファイルは、白の背景に黒で数字が描かれていて、黒画素の画素濃度が 0、白画素の画素濃度が 255 です。この画像を読み込み、`im2double` 関数で倍精度実数に変換すると、黒画素の画素濃度が 0、白画素の画素濃度が 1 の実数になります。識別においては、黒の数字が画像内でどのように分布しているのが重要なので、関心のある黒の画素値を大きく、関心のない背景の画素値を小さく

するようにします。したがって、特徴量の計算では、以下のように画素値を反転させた画像 $\tilde{I}_{x,y}$ を用います。ここで、 $I_{x,y}$ は対象となる画像の座標 (x,y) での画素値です。

$$\tilde{I}_{x,y} = 1 - I_{x,y} \quad (5.1)$$

次に、画像毎の黒画素数の違いが、特徴のスケールに影響しないように、1枚の画像に関して画素値の総和が1になるように画素濃度正規化処理を行います。

$$\hat{I}_{x,y} = \frac{\tilde{I}_{x,y}}{\sum_{x=1}^X \sum_{y=1}^Y \tilde{I}_{x,y}} \quad (5.2)$$

5.3.2 特徴抽出

ここでは、画素値の縦横方向の分散をそれぞれ特徴として用います。分散を計算するために、まず平均を求めます。

平均

$$\begin{aligned} \mu_X &= \sum_{x=1}^X \sum_{y=1}^Y x \hat{I}_{x,y} \\ \mu_Y &= \sum_{x=1}^X \sum_{y=1}^Y y \hat{I}_{x,y} \end{aligned} \quad (5.3)$$

μ_X は横方向の平均、 μ_Y は縦方向の平均を表します。

分散

$$\begin{aligned} \sigma_X^2 &= \sum_{x=1}^X \sum_{y=1}^Y (x - \mu_X)^2 \hat{I}_{x,y} \\ \sigma_Y^2 &= \sum_{x=1}^X \sum_{y=1}^Y (y - \mu_Y)^2 \hat{I}_{x,y} \end{aligned} \quad (5.4)$$

σ_X^2 は横方向の分布の分散、 σ_Y^2 は縦方向の分布の分散を表します。

ここでは、この σ_X^2, σ_Y^2 を標準化したものを特徴ベクトルとします。

実践演習 5-1

上記手順で特徴ベクトルを求めるコードを完成させよ。次に、各クラスの平均ベクトルをプロトタイプとして、最近傍決定則によって全データの識別を行い、性能を評価せよ。

参考コード

```
clear
DATADIR = 'number\';
n = 100; // データ数
y=matrix(repmat([0:9],[10,1]),[100,1]); // 正解ベクトル
```

```
// ファイルを読み込み、特徴量を計算
function result = feature(filename)
    im = im2double(imread(DATADIR+filename));
    [h w] = size(im);

    ...

    result = [vx, vy]
endfunction

// Mainプログラム
// 読み込み・特徴抽出
X=[];
for i=0:9 do
    for j=0:9 do
        X = [X; feature('number'+string(i)+'_'+string(j)+'.pgm')];
    end
end

//標準化
m = mean(X,'r');
s = stdev(X, 'r');
normX = (X - repmat(m,[n,1])) ./ repmat(s,[n,1]);

// グラフ表示
for i=1:10:100 do
    plot2d(normX(i:i+9,1), normX(i:i+9,2), style=-ceil(i/10), rect=[-2,-2,2,2])
end
//凡例
legend(['0','1','2','3','4','5','6','7','8','9'],-1);

// プロトタイプの設定
P=[];
...

// 識別
y2=[];
...
mprintf("result = %f%%\n", 100*(n - nnz(y - y2)) / n);
```
