実践演習 1-1

(ア) `size(P,'c')`

(イ) `:,i`

実践演習 1-2

```
[mindist, ans] = min(sqrt(sum((P-repmat(x,[1,size(P,'c')])).^2,'r')))
```

実践演習 1-3

```
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]';

function feature = feature_extraction(data)
    feature = [];
    for i = 1:size(data, 'c')
        img = matrix(data(:,i), 5, 5)';
        feature = [feature, [detect_line(img), detect_line(img')]'];
    end
endfunction

function val = detect_line(m)
    val = 0;
    for i = 1:size(m,'c')
        if regexp(strcat(string (m(:,i))), '/111/') > 0
            val = val + 1;
         end
    end
endfunction

F = feature_extraction(P);
```

```
x2 = feature_extraction(x);

[mindist, ans] = min(sqrt(sum((F-repmat(x2,[1,size(F,'c')])).^2,'r')));
disp("Ans = "+string(ans-1))
```

実践演習 2-1

(ア) y, x

(イ) y-1:y+1, x-1:x+1

実践演習 2-2

```
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
resultim1 = ones(im);
resultim2 = ones(im);

// フィルタ適用
for y = 2:h-1
    for x = 2:w-1
        resultim1(y, x) = median(im(y-1:y+1, x-1:x+1));
        resultim2(y, x) = mean(im(y-1:y+1, x-1:x+1));
    end
end

// 結果の表示とファイルへの出力
imshow([im, resultim1, resultim2])
imwrite([im, resultim1, resultim2], 'out.png');
```

実践演習 2-3

```
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);
// Sobelフィルタの定義
dx=[-1,0,1; -2,0,2; -1,0,1];
dy=[1,2,1; 0,0,0; -1,-2,-1];

// フィルタ適用
for y = 2:h-1
    for x = 2:w-1
        resultim(y, x) = sqrt(sum(im(y-1:y+1, x-1:x+1) .* dx)^2+..
          sum(im(y-1:y+1, x-1:x+1) .* dy)^2);
    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 2-4

```
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
resultim1 = ones(im);
resultim2 = ones(im);
// Sobelフィルタの定義
dx=[-1,0,1; -2,0,2; -1,0,1];
dy=[1,2,1; 0,0,0; -1,-2,-1];

// フィルタ適用
for y = 2:h-1
    for x = 2:w-1
        resultim1(y, x) = sqrt(sum(im(y-1:y+1, x-1:x+1) .* dx)^2+..
          sum(im(y-1:y+1, x-1:x+1) .* dy)^2);
    end
end

// maxプーリング
for y = 2:h-1
    for x = 2:w-1
        resultim2(y, x) = max(resultim1(y-1:y+1, x-1:x+1));
    end
end

// 結果の表示とファイルへの出力
imshow([im, resultim1, resultim2]);
imwrite([im, resultim1, resultim2], 'out.png');
```

実践演習 3-1

(ア) size

(イ) mean

(ウ) stdev

(エ) m

(オ) s

(カ) pca

実践演習 3-2

```
clear;
M = csvRead('iris.csv');
X = M(:,1:4);
[n d] = size(X);

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

// 主成分分析
[l, f, c] = pca(normX);
plot2d(c(1:50,1), c(1:50,2), style=-3, rect=[-4,-4,4,4])
```

```
plot2d(c(51:100,1), c(51:100,2), style=-4)
plot2d(c(101:150,1), c(101:150,2), style=-5)
```

実践演習 4-1

```
clear;
X = [1.0; 0.5; -0.2; -1.3]; // 学習データ
y = [1 1 2 2]'; // 正解クラス
w = [0.2; 0.3]; // 初期重み
roh = 0.5; // 学習係数
flag = %T;  // 重みに変更があれば TRUE(%T)
[n, d] = size(X);
X = [ones(n,1), X]; // x_0 軸を追加

while flag
  flag = %F;
  for i = 1:n
    x = X(i,:)'
    g = w' * x;
    disp(w');
    if y(i) == 1 & g < 0
      w = w + roh * x;
      flag = %T;
    elseif y(i) == 2 & g > 0
      w = w - roh * x;
      flag = %T;
    end
  end
end
printf("Results: w0=%6.3f, w1=%6.3f\n",w(1), w(2));
```

実践演習 4-2

```
clear;
X = [1 4; 2 3; 4 3; 5 4; 2 1; 3 2; 3 3; 4 1]; // 学習データ
y = [1 1 1 1 2 2 2 2]'; // 正解クラス
k = 3;
x = [3 4]'; // 入力
[n, d] = size(X);

// 入力と学習データとの距離を計算
dist = sqrt(sum((X-repmat(x', [n,1])) .^2, 'c'));

// 上位k個のクラスを取得
[A, B] = gsort(dist, 'g', 'i');
near = y(B(1:k));

// 多数決
[val, ind] = max(members([1,2],near))
printf("Result: class %d", ind);
```

実践演習 4-3

```
clear;
X = [1 4; 2 3; 4 3; 5 4; 2 1; 3 2; 3 3; 4 1]; // 学習データ
y = [1 1 1 1 2 2 2 2]'; // 正解クラス
k = 3;
x = [2.1 3]'; // 入力
[n, d] = size(X);
```

```
// 入力と学習データとの距離を計算
dist = sqrt(sum((X-repmat(x', [n,1])) .^2, 'c'));

// 上位k個のクラスを取得
[A, B] = gsort(dist, 'g', 'i');
near = y(B(1:k));

// 重み付き多数決
v = zeros(1,2)
for i=1:k
     v(near(i)) = v(near(i)) + 1/A(i);
end
[val, ind] = max(v);
printf("Result: class %d", ind);
```