

実践演習 1-1

```
clear;
X = [15 31 71 45 55 62 18 88 50 60];
mprintf("平均点  %4.2f 標準偏差  %4.2f 最高点  %d 最低点  %d", ..
    mean(X),..
    stdev(X),..
    max(X),..
    min(X));
```

実践演習 1-2

```
clear;
X = [175 77;
    168 60;
    180 68;
    160 55;
    165 80];
mprintf("平均身長  %5.2f 平均体重  %4.2f", mean(X(:,1)), mean(X(:,2)));
// mprintf("平均身長  %5.2f 平均体重  %4.2f", mean(X, 'r'));
plot2d(X(:,1), X(:,2), -4, rect=[0,0,200,100])
```

実践演習 2-1

```
(ア) size(P,'c')
(イ) :,i
```

実践演習 2-2

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

実践演習 2-3

```
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,...
         0,1,0,0,0,0,...
         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 3-1

(ア) y, x

(イ) y-1:y+1, x-1:x+1

実践演習 3-2

```

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 3-3

```
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 4-1

- (ア) size
- (イ) mean
- (ウ) stdev
- (エ) m
- (オ) s
- (カ) pca

実践演習 4-2

```
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 5-1

```

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);
    //反転
    im = 1 - im;
    //正規化
    im = im ./ sum(im);
    //平均
    mx = sum(im,'r') * (1:w)';
    my = sum(im,'c')' * (1:h)';
    //分散
    vx = sum(im,'r') * (((1:w) - mx).^2)';
    vy = sum(im,'c')' * (((1:h) - my).^2)';
    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=[];
for i=1:10:100 do
    P = [P, [mean(normX(i:i+9,1)), mean(normX(i:i+9,2))]]';
end

// 識別
y2=[];
for i=1:100 do
    [mindist, ans] = min(sqrt(sum((P-repmat(normX(i,:),[1,size(P,'c')]))).^2,'r')));
    y2 = [y2; ans-1];
end
mprintf("result = %f%%\n", 100*(n-nnz(y-y2))/n);

```

参考コード（行列演算を for ループに置き換えたもの）

```

clear
DATADIR = 'number\';
n = 100; // データ数
y=[];
for i = 0:9
    for j= 1:10
        y = [y; i];
    end
end

```

```

// 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);
    //反転
    for i = 1:h
        for j= 1:w
            im(i,j) = 1 - im(i,j);
        end
    end
    //正規化
    s = sum(im);
    for i = 1:h
        for j= 1:w
            im(i,j) = im(i,j) / s;
        end
    end
    //平均
    mx = 0;
    my = 0;
    for i = 1:h
        for j= 1:w
            mx = mx + j * im(i,j);
            my = my + i * im(i,j);
        end
    end
    //分散
    vx = 0;
    vy = 0;
    for i = 1:h
        for j= 1:w
            vx = vx + (j - mx)^2 * im(i,j);
            vy = vy + (i - my)^2 * im(i,j);
        end
    end
    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=[];
for i = 1:n
    normX = [normX; (X(i,:) - m) ./ s]
end

// グラフ表示
for i=1:10:100
    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=[];
for i=1:10:100
    P = [P, [mean(normX(i:i+9,1)), mean(normX(i:i+9,2))]];
end

// 識別
y2=[];
for i=1:100
    x = normX(i,:);

```

```

        dist = [];
        for j = 1:10
            dist(j) = norm(P(:,j) - x);
        end
        [mindist, ans] = min(dist);
        y2 = [y2; ans-1];
    end
    mprintf("result = %f%%\n", 100*(n-nnz(y-y2))/n);

```

実践演習 6-1

```

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 = [0.8; -0.8]; // 初期重み
roh = 0.02; // 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
mprintf("Results: w0=%6.3f, w1=%6.3f\n", w(1), w(2));

```

実践演習 6-2

```

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 = [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));

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

```

実践演習 7-1

(ア) $\text{inv}(X' * X) * X' * y$ // $w = \text{lsq}(X, y)$; でもよい

実践演習 7-2

```
(ア) w - rho * sum(X .* repmat((X * w - y), [1,2]), 'r')';  
(イ) abs(older - sqrrerr);
```

実践演習 7-3

図 8.4 のデータでは、 $w_0 = -0.44, w_1 = 0.00, w_2 = 0.22$ となり、識別面の式は $0.22x_2 - 0.44 = 0 \Leftrightarrow x_2 = 2$ となります。また、点 $(3, 3.9)$ をクラス ω_2 に加えると、 $w_0 = -0.57, w_1 = -0.00, w_2 = 0.21$ から、識別面は $x_2 = 2.71$ となり、点 $(3, 3.9)$ を誤識別します。

実践演習 8-1

```
Kernel used:  
Poly Kernel: K(x,y) = <x,y>^3.0  
Classifier for classes: 1, -1  
BinarySMO  
- 0.2073 * <4 3 > * X]  
+ 0.4911 * <3 3 > * X]  
+ 0.0359 * <4 1 > * X]  
- 0.3197 * <2 3 > * X]  
+ 14.661
```

実践演習 8-2

```
--> function y=f(x), y= - 0.2073 * ([4 3] * x)^3 + 0.4911 * ([3 3] * x)^3 +..  
0.0359 * ([4 1] * x)^3 - 0.3197 * ([2 3] * x)^3 + 14.661, endfunction  
-->f([1 4]')  
ans =  
- 35.8533  
-->f([2 1]')  
ans =  
13.2706  
-->f([3 4]')  
ans =  
- 20.4211  
-->f([3 1]')  
ans =  
9.4553
```

実践演習 8-3

実践演習 8-1 のデータファイルにおいて、class 属性が取り得る値を -1, 0, 1 のようにして、いくつかのデータのクラス属性の値を変更します。SMO を実行すると、Classifier output 領域に以下のように表示されるので、1 対 1 の分類器をクラス対毎に作っていることがわかります。

```
Classifier for classes: 1, 0  
BinarySMO
```

実践演習 9-1

デフォルトのパラメータでは1つのデータを誤識別します。trainingTime を 5000 に増やすと誤識別が0となります。

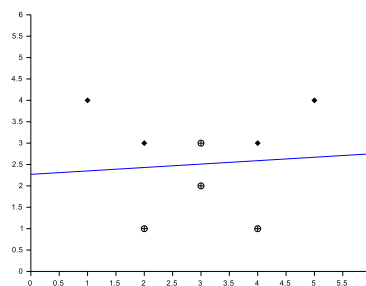
実践演習 9-2

デフォルトのパラメータでは3つのデータを誤識別します。hiddenLayers を 6, 6 (2層 6 ユニット + 6 ユニット)、trainingTime を 5000 に増やすと誤識別が0となります。

実践演習 9-3

hiddenLayers を 0 にすると単層パーセプトロンになります。学習結果は以下のようになるので、Node0-Node1 の式を求めると $x_2 = 0.08x_1 + 2.27$ になります。

```
Sigmoid Node 0
  Inputs  Weights
Threshold -5.984053487366797
Attrib x1 -0.1996798919110183
Attrib x2  2.6356689153702564
Sigmoid Node 1
  Inputs  Weights
Threshold  5.983386250908053
Attrib x1  0.19973332566782634
Attrib x2 -2.6354901242127253
```



実践演習 10-1

```
clear; clf();
function z=normal(x,y)
    z = 1/((2*pi) * det(S)^(0.5))...
        * exp(-0.5 * ([x;y]-m)' * inv(S) * ([x;y]-m));
endfunction

X1 = [3 4; 3 8; 2 6; 4 6]; // クラス1のデータ
X2 = [3 0; 1 -2; 5 -2; 3 -4]; // クラス2のデータ
// 各クラスの平均ベクトルと共分散行列
```



```

m1 = mean(X1, 'r')';
m2 = mean(X2, 'r')';
S1 = cov(X1);
S2 = cov(X2);

x = [-1:0.5:8]; y = [-8:0.5:12]; // プロットする範囲
m = m1; S = S1;
fplot3d(x,y,normal,flag=[2,2,4]);
m = m2; S = S2;
fplot3d(x,y,normal,flag=[4,2,4],alpha=89.5,theta=-62);

```

実践演習 10-2

```

clear;
c = 10; // クラス数
function z=normal(x,y)
    z = 1/((2*pi)*det(S)^(0.5))...
        *exp(-0.5*([x,y]-m)*inv(S)*([x,y]-m)');
endfunction

function result=recog(x)
    score = [];
    for i = 1:c
        m = mv(:,i)';
        S = cm(:,i,i);
        score = [score, normal(x(1), x(2))];
    end
    [high arg] = max(score);
    result = arg - 1;
endfunction

M = csvRead('numbers.csv');
X = M(:,1:2);
y = M(:,3);
[n d] = size(X);

// 識別関数を求める
mv = zeros(d,c);
cm = zeros(d,d,c);
for i=1:10
    data = X((i-1)*10+1:i*10, :);
    mv(:,i) = mean(data,'r')';
    cm(:,i,i) = cov(data);
end

// 識別
y2=[];
for i=1:100 do
    y2 = [y2; recog(X(i,:))];
end
fprintf("result = %f%%\n", 100*(n-nnz(y-y2))/n);

```

実践演習 11-1

```

// クラス内分散・クラス間分散比の計算
function[result] = wcbc(va, vb)
    wc = 0;
    bc = 0;
    for i=1:10:100
        Xi = [va(i:i+9), vb(i:i+9)];
        mi = mean(Xi, 'r');
        wc = wc + sum((Xi - repmat(mi,[10,1])).^2);
        bc = bc + 10 * sum(mi.^2);
    end
    result = [wc, bc];
endfunction

```

```

    end
    result = bc / wc;
endfunction

```

実践演習 11-2

```

// 定性的評価
[a b] = gsort(J);
for i = 1:3
    f1 = modulo(b(i),d);
    f2 = ceil(b(i)/d);
    mprintf("Best %d: f1 = %d, f2 = %d\n", i, f1, f2);
    subplot(3,1,i);
    for j=1:10:n do
        plot2d(normX(j:j+9,f1), normX(j:j+9,f2), style=-ceil(j/10), rect=[-2,-2,2,2])
    end
    legend(['0','1','2','3','4','5','6','7','8','9'],-1);
end

```

実践演習 11-3

```

// 1-NN 法による認識率計算
function result = kNN(va, vb)
    y2 = [];
    for i=1:n
        x = [va(i) vb(i)]; // i 番目のデータ
        proto = [va(1:i-1) vb(1:i-1); va(i+1:$) vb(i+1:$)]; // i 番目以外のデータ
        [a b] = min(sqrt(sum((proto-repmat(x, [n-1,1])).^2,'c')));
        y2 = [y2; y(b)] ;
    end
    result = 100*((n-1) - nnz(y-y2)) / (n-1);
endfunction

```