

パターン認識演習 課題6

課題6 フィッシャーの判別分析

課題6.1 プログラムの作成

フィッシャーの判別分析を行うプログラムを作成せよ。
以下にプログラムの流れの例を示す。

1. クラス ω_1 に属するデータをファイルから読み込む
2. クラス ω_2 に属するデータをファイルから読み込む
3. クラス ω_1 に属するデータの平均 m_1 を求める
4. クラス ω_2 に属するデータの平均 m_2 を求める
5. クラス ω_1 の変動行列 S_1 を求める
6. クラス ω_2 の変動行列 S_2 を求める
7. クラス内変動行列 S_w を求める
8. S_w の逆行列 S_w^{-1} を求める
9. 変換ベクトル w (教科書では、行列 A) を求める
10. 変換ベクトル w を正規化
11. クラス ω_1 のデータに対する変換値 y を求め、ファイルに出力
12. クラス ω_2 のデータに対する変換値 y を求め、ファイルに出力

なお、逆行列は以下の関数を利用してよい。

[逆行列を求める関数inverse](#)

逆行列を求める関数inverseの使用例

```
#include <stdio.h>
#include <stdlib.h>

main()
{
    int i, j, k, n;
    double **mat, **inv;

    /*次元入力*/
    printf( "Dimension=" );
    scanf( "%d", &n);
    /* メモリ確保 */
    mat = (double**)malloc(n*sizeof(double*));
    inv = (double**)malloc(n*sizeof(double*));
    for( i = 0; i < n; i++ ){
        mat[i] = (double*)malloc(n*sizeof(double));
        inv[i] = (double*)malloc(n*sizeof(double));
    }

    /*行列入力*/
    printf( "Matrix=%zn" );
    for( i = 0; i < n; i++ ){
        for( j = 0; j < n; j++ ){
```

```

        scanf( "%lf", &mat[i][j] );
    }
}

/*逆行列*/
inverse(mat, inv, n);

/*逆行列出力*/
printf( "Inverse Matrix=%zn");
for( i = 0; i < n; i++ ){
    for( j = 0; j < n; j++ ){
        printf( "%g ", inv[i][j] );
    }
    printf("%zn" );
}
}

```

```

Dimension=2
Matrix=
1 2
2 3
Inverse Matrix=
-3 2
2 -1

```

入力ファイル例

```

-----cls1.dat-----
4 2 //データ数 次元数
1 2 //x1
4 4 //x2
2 11 //x3
5 11 //x4
-----cls2.dat-----
4 2 //データ数 次元数
5 2 //x5
6 4 //x6
6 8 //x7
11 10 //x8

```

出力ファイル例

```

-----cls1.out-----
-0.234613 1 //y1 クラス
-2.34205 1 //y2 クラス
1.98707 1 //y3 クラス
-0.822167 1 //y4 クラス
-----cls2.out-----
-3.98027 2 //y5 クラス
-4.21488 2 //y6 クラス
-2.81128 2 //y7 クラス
-6.79154 2 //y8 クラス

```

課題6.2 手書き文字データの判別

手書き文字データのGlucksmanの特徴を用いて、二つの文字クラスを判別分析せよ。

ただし、このとき16次元のGlucksmanの特徴を用いる物とする。

16次元のGlucksmanの特徴を求めるには、前回の課題で配布したglucksman.hで定義されている定数DIMを2にすればよい。具体的には、glucksman.hを

```
#ifndef _GLUCKSMAN

/*16次元のGlucksmanの特徴を求める場合は2*/
/*81次元のGlucksmanの特徴を求める場合は3*/
/*コメントアウト*/
/*#define DIM 3*/
/*追加*/
#define DIM 2

#define ELM_SIZE (DIM*DIM*DIM*DIM)

typedef struct {
    int **data;
    int width;
    int height;
} MojiData;

void getGlucksmanVector(MojiData *mojiData, int data[]);

#endif /*_GLUCKSMAN*/
```

とすればよい。

用いるデータは、learndataとし、一クラス100個のデータを用いる。

また、データはGlucksmanの特徴に変換し利用する。

比較は、すべての組み合わせについて行い、それぞれのクラス内クラス間変動比(教科書P.117式6.97~98)を求めよ。

求めたクラス内クラス間変動比は表にまとめること。

なお、

- 0と7
- 0と8
- 2と3
- 4と9
- 8と9

については、判別結果の図をレポートに載せること。

次元の圧縮

Glucksmanの特徴では、判別しようとするすべての文字において、同一の値をとる要素が存在することがある。

このような要素は、判別に利用できないばかりでなく、このような要素を利用することで、逆行列 S_w^{-1} を求めることができなくなる。

そのため、このような特徴量として役に立たない要素を排除する必要がある。

すなわち、すべての文字において0となる要素をのぞいた特徴量を用いて比較を行うこと。

次元圧縮の例

```
0 0 1 2 5
0 3 1 2 4
0 4 3 4 8
```

という特徴量があったならば、最初の要素はすべての文字で値が0であるため、特徴量として利用できない。そこで、

```
0 1 2 5
3 1 2 4
4 3 4 8
```

と該当する要素を削除する。

Gnuplotの利用方法

課題6.2の結果を図示の際には、以下のようにgnuplotを利用してよい。

gnuplotの使い方:

ファイル"cls1.out", "cls2.out"をy軸の範囲0-3でプロット。

同じ図をepsファイルとして、"fisher.eps"に出力する

```
$ gnuplot

G N U P L O T
Version 4.0 patchlevel 0
last modified Thu Apr 15 14:44:22 CEST 2004
System: Linux 2.4.27-2-386

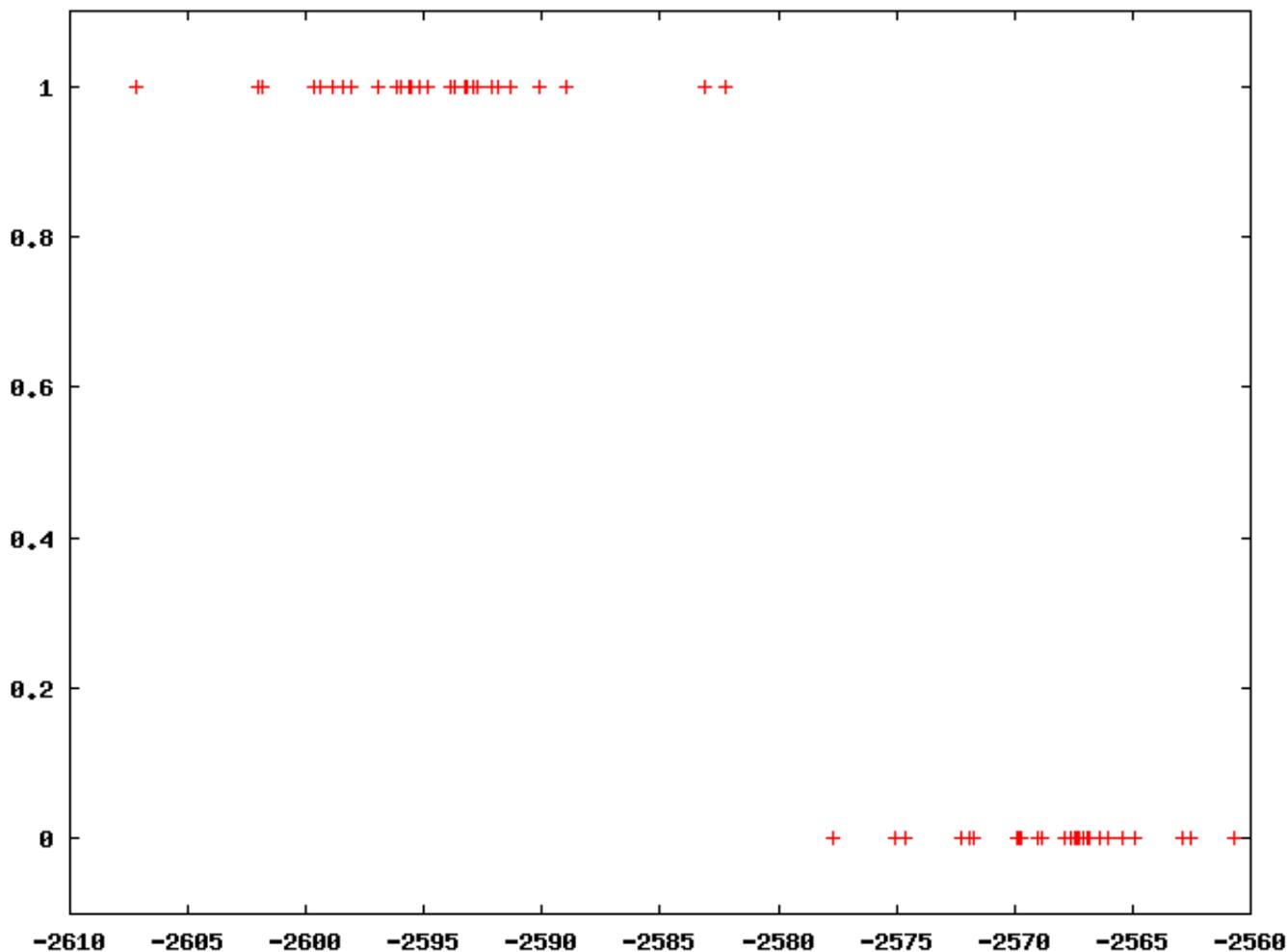
Copyright (C) 1986 - 1993, 1998, 2004
Thomas Williams, Colin Kelley and many others

This is gnuplot version 4.0. Please refer to the documentation
for command syntax changes. The old syntax will be accepted
throughout the 4.0 series, but all save files use the new syntax.

Type `help` to access the on-line reference manual.
The gnuplot FAQ is available from
    http://www.gnuplot.info/faq/

Send comments and requests for help to
    <gnuplot-info@lists.sourceforge.net>
Send bugs, suggestions and mods to
    <gnuplot-bugs@lists.sourceforge.net>

Terminal type set to 'x11'
gnuplot> plot"cls1.out" u 1:$0,"cls2.out"u 1:$0 //cls1.outとcls2.outをプロット
gnuplot> set terminal png //出力形式をpngファイルにする
gnuplot> set output "fisher.png" //出力ファイルをfisher.pngにする
gnuplot> replot //いざ出力
gnuplot> quit //gnuplot終了
$ ls
cls1.out cls2.out fisher.png ←できた
$
```



プロット結果:fisher.png

クラス ω_1 のデータとクラス ω_2 のデータが判別出来ていることが確認できる。

課題6.3 変換ベクトルに関する考察

課題6.2で得られた変換ベクトル \mathbf{w} について、特に特徴的な変換ベクトルを1つあげ考察せよ。

課題6.4 筆者の判別分析

実験で取得した「5」について、自分で書いた文字と他人が書いた文字で判別可能かどうか確認せよ。

各自が書いた5のデータについては、[こちら](#)からダウンロードできる。

解凍方法は以下の通り。

```
$ tar xzf five.tgz
```

解凍が終了すると、

学籍番号/5-00.dat

学籍番号/5-01.dat

...

学籍番号/5-29.dat

というデータが得られる。自分の学籍番号の直下に入っているデータは、自分で書いた文字である。

自分で書いた5と他人が書いた5について判別分析可能かどうか調べよ。

なお、同一文字を処理するため、次元を圧縮しなければ逆行列は求められない可能性が高いので、注意すること。

応用課題6.5 いろいろなデータの判別分析

時間があまっている人はやってください

興味がある分野のデータを用いた判別分析せよ。

もし、判別が困難なデータだった場合は、その理由を考察すること。