パタン認識特論
第1回レポート課題

3G150131　松井 健人

2015 年 5 月 20 日

1　課題

教科書「わかりやすいパターン認識（オーム社）」の「2.3 パーセプトロンの学習規則」の内容に従って，1 次元データ（図 2.4）に対するパーセプトロンの計算を実行し，23 ページの図 2.7 を作図せよ．計算に用いる言語は Java か C とする．

提出物：
①. 作成したプログラムソースコードを印刷したもの．
②. 作図結果（ρ の値を 1.2 と 2.0，3.6 に設定した係数ベクトルの変化パタンを 3 種）．
※提出物の様式は自由．但し，学籍番号と氏名は忘れずに !!
提出期限：2015 年 5 月 20 日．提出先：恵喜館の片桐のメールボックス．

2　パーセプトロンの学習規則

パーセプトロンの学習規則とは，線形識別関数の重みを学習によって決定する方法である．パーセプトロンの学習規則の手順を以下に示す．

(1) 重みベクトル $\boldsymbol{w}$ の初期値を適当に設定する．

(2) χ の中から学習パターンを 1 つ選ぶ．

(3) 識別関数 $g(x) = \boldsymbol{w}^T\boldsymbol{x}$ によって識別を行い，正しく識別できなかった場合のみ次のように重みベクトル $\boldsymbol{w}$ を修正し，新しい重みベクトル $\boldsymbol{w'}$ を作る．

$$\boldsymbol{w'} = \boldsymbol{w} + \rho\cdot\boldsymbol{x} \qquad (\omega_1\text{のパターンに対して } g(x) \leq 0 \text{ となったとき}) \tag{1}$$

$$\boldsymbol{w'} = \boldsymbol{w} - \rho\cdot\boldsymbol{x} \qquad (\omega_2\text{のパターンに対して } g(x) \geq 0 \text{ となったとき}) \tag{2}$$

(4) 上の処理 (2)，(3) を χ の全パターンに対して繰り返す．

(5) χ の全パターンを正しく識別できたら終了．誤りがあるときは (2) に戻る．

3　作成したプログラムソースコード

2 章で示したパーセプトロンの学習規則を Java を用いて作成した．2 つのクラス ω_1，ω_2 が存在し，それぞれに学習パターンが 3 つずつ属している．ω_1 に属する学習パターンは，$(x_1, x_2, x_3) = (1.2, 0.2, -0.2)$ であり，ω_2 に属する学習パターンは，$(x_4, x_5, x_6) = (-0.5, -1.0, -1.5)$ である．この学習パターンを基に，ρ の値を 1.2，2.0，3.6 に設定し，重みベクトル $\boldsymbol{w}$ をそれぞれ $(w_1, w_0) = (2, -7)$，$(w_1, w_0) = (5, 11)$，$(w_1, w_0) = (2, -7)$ として計算を行った．ソースコードをリスト 1 に示す．また，出力結果をリスト 2 に示す．

リスト 1: プログラムソースコード

```
1   import java.text.DecimalFormat;
2   import java.util.ArrayList;
3
4   /**
5    * パーセプトロンの収束定理
6    *
7    */
8   public class PerceptronConvergenceTheorem {
9     /* クラスω１の学習パターン：(x1, x2, x3) = (1.2, 0.2, -0.2)
10     * クラスω２の学習パターン：(x1, x2, x3) = (-0.5, -1.0, -1.5) */
11
12    final double X[] = { 1.2, 0.2, -0.2, -0.5, -1.0, -1.5 };  //X:学習パターン
13    final double THRESHHOLD = -2.0 / 5.0;                     //クラスω１とω２を識別する際の閾値
14    DecimalFormat df = new DecimalFormat("0.00");             //表示する桁数を設定
15
16    public static void main(String[] args) {
17      new PerceptronConvergenceTheorem();
18    }
19
20    public PerceptronConvergenceTheorem(){
21      //パーセプトロンの学習規則
22      PerceptronLearningRule(1.2, -7, 2);      //ρの値＝ 1.2，重みベクトルｗ (w0, w1) = (-7, 2)
23      PerceptronLearningRule(2.0, 11, 5);      //ρの値＝ 2.0，重みベクトルｗ (w0, w1) = (11, 5)
24      PerceptronLearningRule(3.6, -7, 2);      //ρの値＝ 3.6，重みベクトルｗ (w0, w1) = (-7, 2)
25    }
26
27    /*パーセプトロンの学習規則 (正の定数ρ，重みベクトルｗのｗ０，重みベクトルｗのｗ１)*/
28    public void PerceptronLearningRule(double p, double w0, double w1){
29      double g;                      //識別関数，g(x) = w0 + w1 * x
30      int counter;                   //全パターンを正しく識別したかを判断する変数
31      ArrayList<Double> temp_w0
32        = new ArrayList<Double>();  //新しく重みベクトルｗを作成した際にｗ０の履歴を保存するリスト
33      ArrayList<Double> temp_w1
34        = new ArrayList<Double>();  //新しく重みベクトルｗを生成した際にｗ１の履歴を保存するリスト
35
36      System.out.println("ρの値＝"+ p +", 重みベクトルｗ＝（"+ df.format(w0) +", "+ df.format(w1) +")");
37      System.out.println("学習パターン　識別関数 g(x)　w0　w1");
38
39      temp_w0.add(w0);               //重みベクトルｗのｗ０の初期値を保存
40      temp_w1.add(w1);               //重みベクトルｗのｗ１の初期値を保存
41
42      /*全ての学習パターンにおいて識別を行う*/
43      while(true){                   //全ての学習パターンにおいて正しく識別できるまでループ
44        counter = 0;                 //カウンターを初期化
45        for(int i = 0; i < X.length; i++){    //学習パターンの数だけループ
46
47          g = w0 + w1 * X[i];        //識別関数 g(x) を算出
```

```
48
49          System.out.println(X[i] +"  "+ df.format(g) +"  "+ df.format(w0) +"  "+ df.format(w1));
50
51          /* 識別関数 g(x) > 0 のとき，その学習パターンはクラスω１であると識別される
52           * 識別関数 g(x) < 0 のとき，その学習パターンはクラスω２であると識別される * */
53
54            if(X[i] > THRESHHOLD){  //学習パターンが閾値以上であり，クラスω１であるか識別する場合
55            if(g > 0){                //学習パターンがクラスω１であると識別された場合
56              counter++;                //正しく識別されたため，counter をインクリメントする
57            }
58            else{                     //学習パターンがクラスω２であると識別された場合
59
60              /* 正しく識別されなかったため，新たな重みベクトルｗ’を作成する
61               *クラスω１の学習パターンに対してクラスω２であると認識した場合
62               *新しい重みベクトルｗ’＝ 重みベクトルｗ ＋ 正の定数ρ となる*/
63
64              w0 += p;                  //新しい重みベクトルｗ’のｗ’０を作成する
65              w1 += p * X[i];           //新しい重みベクトルｗ’のｗ’１を作成する
66              temp_w0.add(w0);          //新しい重みベクトルｗ’のｗ’０を保存する
67              temp_w1.add(w1);          //新しい重みベクトルｗ’のｗ’１を保存する
68            }
69          }
70
71          else{                     //学習パターンが閾値以下であり，クラスω２である識別する場合
72            if(g < 0){                //学習パターンがクラスω２であると識別された場合
73              counter++;                //正しく識別されたため，counter をインクリメントする
74            }
75            else{                     //学習パターンがクラスω１であると識別された場合
76
77            /*正しく識別されなかったため，新たな重みベクトルｗ’を作成する
78             *クラスω２の学習パターンに対してクラスω１であると認識した場合
79             *新しい重みベクトルｗ’ ＝ 重みベクトルｗ ー 正の定数ρ となる*/
80
81              w0 -= p;                  //新しい重みベクトルｗ’のｗ’０を作成する
82              w1 -= p * X[i];           //新しい重みベクトルｗ’のｗ’１を作成する
83              temp_w0.add(w0);          //新しい重みベクトルｗ’のｗ’０を保存する
84              temp_w1.add(w1);          //新しい重みベクトルｗ’のｗ’１を保存する
85            }
86          }
87        }
88
89        if(counter == X.length){      //全ての学習パターンにおいて正しく識別された場合
90          break;                      //ループから抜ける
91        }
92      }
93
94      /*重みベクトルの初期値と新しく作成した重みベクトルを表示*/
95      System.out.println("学習パターン  w0  w1");
96      for(int i = 0; i < temp_w0.size(); i++){    //新しく作成した重みベクトルの数だけループ
97        System.out.println(df.format(temp_w0.get(i)) +"  "+ df.format(temp_w1.get(i)));
98      }
99    }
100 }
```

リスト 2: 出力結果

```
ρの値＝ 1.2，重みベクトルw＝（-7.00，2.00）
学習パターン　識別関数 g(x)　　w0　　w1
1.2　　-4.60　　-7.00　2.00
0.2　　-5.11　　-5.80　3.44
-0.2　　-5.34　　-4.60　3.68
-0.5　　-5.12　　-3.40　3.44
-1.0　　-6.84　　-3.40　3.44
-1.5　　-8.56　　-3.40　3.44
1.2　　0.73　　-3.40　3.44
0.2　　-2.71　　-3.40　3.44
-0.2　　-2.94　　-2.20　3.68
-0.5　　-2.72　　-1.00　3.44
-1.0　　-4.44　　-1.00　3.44
-1.5　　-6.16　　-1.00　3.44
1.2　　3.13　　-1.00　3.44
0.2　　-0.31　　-1.00　3.44
-0.2　　-0.54　　0.20　3.68
-0.5　　-0.32　　1.40　3.44
-1.0　　-2.04　　1.40　3.44
-1.5　　-3.76　　1.40　3.44
1.2　　5.53　　1.40　3.44
0.2　　2.09　　1.40　3.44
-0.2　　0.71　　1.40　3.44
-0.5　　-0.32　　1.40　3.44
-1.0　　-2.04　　1.40　3.44
-1.5　　-3.76　　1.40　3.44
重みベクトルw
w0　　w1
-7.00　2.00
-5.80　3.44
-4.60　3.68
-3.40　3.44
-2.20　3.68
-1.00　3.44
0.20　3.68
1.40　3.44

ρの値＝ 2.0，重みベクトルw＝（11.00，5.00）
学習パターン　識別関数 g(x)　　w0　　w1
1.2　　17.00　　11.00　5.00
0.2　　12.00　　11.00　5.00
-0.2　　10.00　　11.00　5.00
-0.5　　8.50　　11.00　5.00
-1.0　　3.00　　9.00　6.00
-1.5　　-5.00　　7.00　8.00
1.2　　16.60　　7.00　8.00
0.2　　8.60　　7.00　8.00
-0.2　　5.40　　7.00　8.00
-0.5　　3.00　　7.00　8.00
-1.0　　-4.00　　5.00　9.00
-1.5　　-8.50　　5.00　9.00
1.2　　15.80　　5.00　9.00
0.2　　6.80　　5.00　9.00
-0.2　　3.20　　5.00　9.00
-0.5　　0.50　　5.00　9.00
-1.0　　-7.00　　3.00　10.00
-1.5　　-12.00　　3.00　10.00
1.2　　15.00　　3.00　10.00
```

```
0.2             5.00            3.00    10.00
-0.2            1.00            3.00    10.00
-0.5            -2.00           3.00    10.00
-1.0            -7.00           3.00    10.00
-1.5            -12.00          3.00    10.00
重みベクトルw
w0      w1
11.00   5.00
9.00    6.00
7.00    8.00
5.00    9.00
3.00    10.00

ρの値＝3.6，重みベクトルw＝（-7.00，2.00）
学習パターン　識別関数g(x)        w0      w1
1.2             -4.60           -7.00   2.00
0.2             -2.14           -3.40   6.32
-0.2            -1.21           0.20    7.04
-0.5            0.64            3.80    6.32
-1.0            -7.92           0.20    8.12
-1.5            -11.98          0.20    8.12
1.2             9.94            0.20    8.12
0.2             1.82            0.20    8.12
-0.2            -1.42           0.20    8.12
-0.5            0.10            3.80    7.40
-1.0            -9.00           0.20    9.20
-1.5            -13.60          0.20    9.20
1.2             11.24           0.20    9.20
0.2             2.04            0.20    9.20
-0.2            -1.64           0.20    9.20
-0.5            -0.44           3.80    8.48
-1.0            -4.68           3.80    8.48
-1.5            -8.92           3.80    8.48
1.2             13.98           3.80    8.48
0.2             5.50            3.80    8.48
-0.2            2.10            3.80    8.48
-0.5            -0.44           3.80    8.48
-1.0            -4.68           3.80    8.48
-1.5            -8.92           3.80    8.48
重みベクトルw
w0      w1
-7.00   2.00
-3.40   6.32
0.20    7.04
3.80    6.32
0.20    8.12
3.80    7.40
0.20    9.20
3.80    8.48
```

4　作図結果

3 章に示したプログラムにより計算した重みベクトルの遷移を Excel を用いて作図した結果を図 1 に示す.

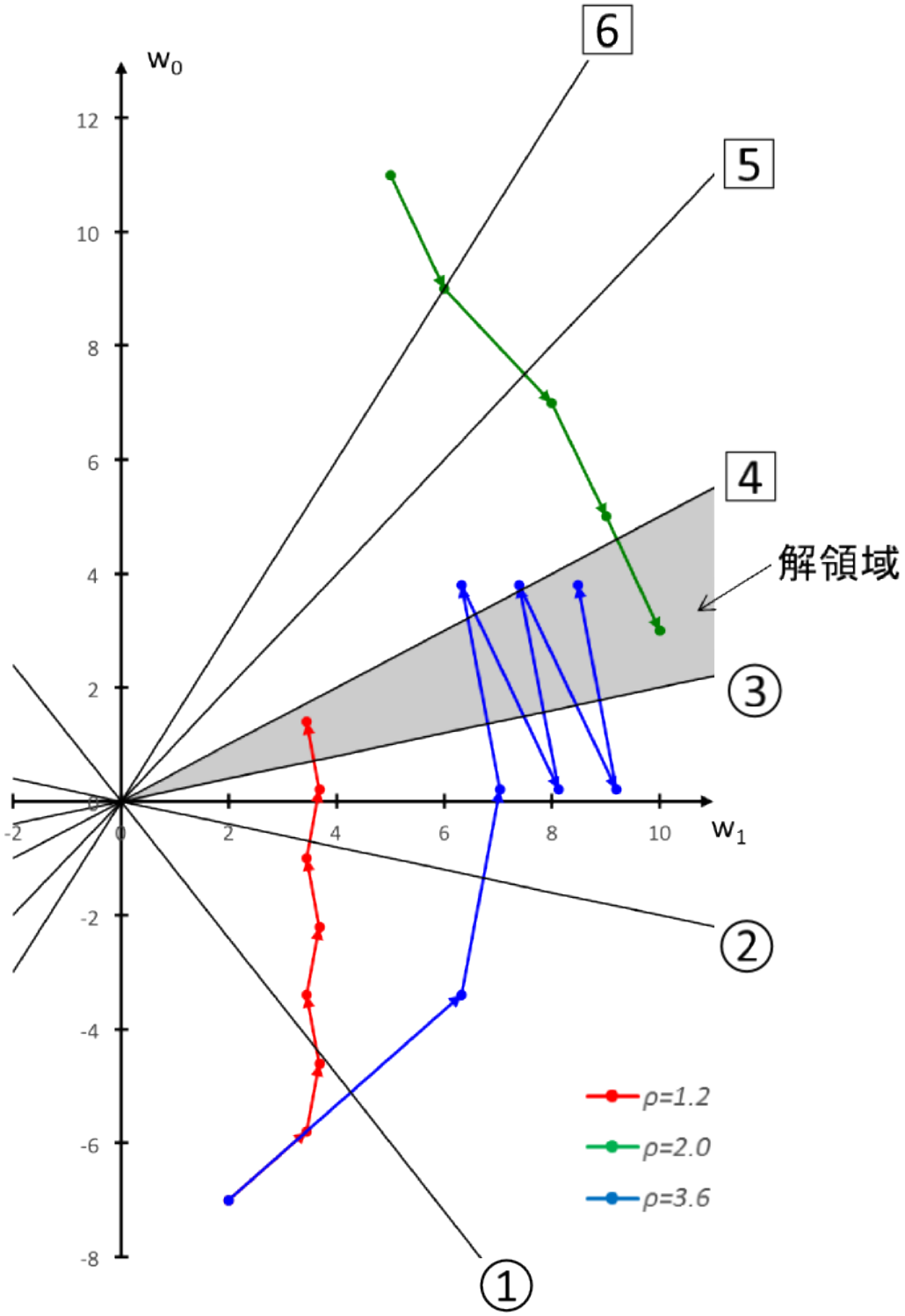

図 1: 重みベクトルの遷移