パターン認識 – 課題 3

GENG Haopeng 611710008
Email: kevingenghaopeng@gmail.com

Department of Intelligent Systems, Nagoya University

2018 年 6 月 9 日

概要

今回の課題は、線形分離可能なパターンをパーセプトロンによるクラスタリングである。さらに、パーセプトロンの原理を理解した上で、実際問題を解決することである。そのほか、課題のソースコードが既に Github Repository に掲載されるため、参照あるいは実行することは可能である。

1 パーセプトロンによる学習 (基本編)

1.1 実験理論およびアルゴリズム

この課題は、パーセプトロンの収束定理を利用し、重みベクトルを修正しつつ、線形分離可能なパターンであれば、解領域内の重みベクトルに到達する。いわゆる、全ての学習パターンが正しく識別される。アルゴリズム [1] の手順は以下のように表す：

1 重みベクトル $\boldsymbol{w}$ の初期値を決める
2 パターン集合 $\mathcal{X}$ から順番で一つのパターン ω_i を選ぶ
3 識別関数

$$g(\boldsymbol{x}) = \boldsymbol{w}^t \boldsymbol{x}$$

によって識別を行い。正しく識別できなかった場合、次のように重みベクトルを修正する。

$$\boldsymbol{w}' = \boldsymbol{w} + \rho \boldsymbol{x} (\omega_i \in \mathcal{X}, g_i(\boldsymbol{x}) \neq \max\{g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \cdots, g_n(\boldsymbol{x})\})$$

$$\boldsymbol{w}' = \boldsymbol{w} - \rho \boldsymbol{x} (\omega_j \in \mathcal{X}, g_j(\boldsymbol{x}) = \max\{g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \cdots, g_n(\boldsymbol{x})\})$$

4 上の処理 [2], [3] 全てのパターンに対して繰り返す
5 $\forall \omega_i \in \mathcal{X}$, 正しく識別できたら終了、誤りがあるときは [2] に戻る

ただし、最大値が複数個出現する場合、計算順で最初に出た重みのみ修正する。

1.2 プログラムに工夫した部分

誤り識別するとき、重みベクトルの修正は以下順番で処理する。

1 全ての重みベクトルに対応する信号量を判断する、すべて「修正済み」になる場合、ステップ 5 へ
2 該当パターンにおいて、識別関数を計算する
    2.1 誤識別の場合、修正式による重みの修正

2.2 正しい識別の場合、収束信号量を「修正済み」に設定

3 次のパターンを導入

4 全ての重みが修正終了

ソースコード 1　correct weight vector

```
/* calculate until converge, while converge condition is all pattern have become stable */
 while(conv(flag, LEARNING_NUM)){
   if(p >= LEARNING_NUM) p = p % LEARNING_NUM;
   for(j = 0; j < CLU_NUM; j++){
     eval[j] = multi(p_arr[p].data, w[j], CLU_NUM);
       }
   /* Corr is pattern's true Class */
   /* Err is Evaluated Class, if condition is met , err == -1 */
   int corr = p_arr[p].pclass;
   int err = judge_max(eval, CLU_NUM, corr);
   printf("Right␣Class␣=␣%d,Error␣Class␣=␣%d\n",corr,err);
   if(err != -1){
     flag[p] = 1;
     for(j = 0; j < W_DIM; j++){
       w[corr][j] += rho * p_arr[p].data[j];
       w[err][j] -= rho * p_arr[p].data[j];
     }
   }
   /* Weight Condition is met, change flag to let result converge */
   else{
     flag[p] = 0;
   }
   printf("========\n");
   for(i = 0; i < CLU_NUM; i++){
     for(j = 0; j < W_DIM; j++){
       printf("%.0f␣",w[i][j]);
     }
     printf("\n");
   }
   p++;
 }
```

1.3　プログラム実行例

1.3.1　重みベクトルの修正

与えられたパターンと初期重みを用い、以下のように出力される。ただし、20 回の繰り返し演算を略し、結果のみを表す。

```
$./pl learning_data.list init_weight.dat weight.dat
.
.
.
Evaluation of Pattern 1 by g(0) : 13.000000
Evaluation of Pattern 1 by g(1) : 9.000000
Evaluation of Pattern 1 by g(2) : 12.000000
Right Class = 0,Error Class = -1
========
9 2 0
2 1 5
-2 6 2
```

出力結果について、

1 各評価関数の値

2 所属すべきクラス、識別関数による識別結果

3 修正された重みベクトル

という順番で示され、ゆえに、合計 20 回の修正で、学習結果である重みベクトルは $\boldsymbol{W} = \begin{bmatrix} 9 & 2 & 0 \\ 2 & 1 & 5 \\ -2 & 6 & 2 \end{bmatrix}$ になった。

1.3.2 未知パターン識別

上記の重みベクトルを用い、未知パターンの識別を行った。

```
$./pl_rec weight.dat unknown.dat
Value of g[0]: 13.000000
Value of g[1]: 14.000000
Value of g[2]: 14.000000
recog result == -1
```

実験結果から、識別関数の最大値が複数個（g[1], g[2]）あるため、識別できない結果になることがわかった。

1.4 考察

1.4.1 初期ベクトルの変化による影響

$\boldsymbol{W_{init}}$ をランダムに 500 個を選び、識別結果を評価した結果は表 1 で示す。

表 1　500 個の初期重みベクトルの実験結果 ($w_{ij} \in [-5, 5], w_{ij} \in \boldsymbol{Z}$)

識別結果	w_1	w_2	w_3	Unrec
識別数	131	84	180	105
比率	26.2%	16.8%	36.0%	21.0%

よって、正しい識別率は 26.2% になった。つまり、初期重みは識別結果に大きな影響を与え得ることがわかった。

1.4.2 定数 ρ の変化による影響

定数 ρ を 0.1 から 1.5 まで、刻み幅を 0.1、および $\rho = 2.0, 3.0, 4.0, 5.0, 6.0$ に設定し実験した結果は表 2 のように表す。

表 2　ρ の変化による実験結果

ρ	識別結果	ρ	識別結果	ρ	識別結果	ρ	識別結果	識別率
0.1	√	0.6	√	1.1	Err(w_2)	2.0	√	75.0%
0.2	√	0.7	√	1.2	√	3.0	Unrec	
0.3	√	0.8	√	1.3	√	4.0	√	
0.4	√	0.9	√	1.4	√	5.0	√	
0.5	Unrec	1.0	Unrec	1.5	Unrec	6.0	√	

実験結果により、$\rho = 0.5, 1.0, 1.5, 3.0$ の時、識別できない場合は多いことがわかり、初期重みとパターンの値は整数であるは原因であると考える。なお、修正係数が大きくなったら識別結果が悪くなると想定したが、実験結果としてそれほど影響していないことがわかった。

2　パーセプトロンによる学習 (応用編)

2.1　実験目的

上記の実験に基づき、日本各地の座標を学習パターンとし、パーセプトロンによる学習で地域の天気を予測することを実験した。ただし、処理しやすいため、初期重みを一定範囲内にランダムに設定し、重み修正幅 $\rho = 1.0$ と設定し、各クラスターを 晴れ $= 0,$ 曇り $= 1,$ 雨 $= 2$ と設定する。

2.2　プログラム実行例および実行結果

pattern7-5.dat を例として、以下のように出力される。

```
$./pl weather.list wea_init_weight.dat weather_weight.dat 1
.
.
.
Evaluation of Pattern 2 by g(0) : 10.110000
Evaluation of Pattern 2 by g(1) : 18.690000
Evaluation of Pattern 2 by g(2) : -4.000000
Right Class = 1,Error Class = -1
========
0.000000 2.500000 8.700000
3.000000 5.000000 2.300000
2.000000 -1.500000 -5.000000

$./pl_rec weather_weight.dat unk_weather.dat
Value of g[0]: -5.480000
Value of g[1]: -1.920000
Value of g[2]: 5.200000
recog result == 2
```

出力結果について、得られた修正済みの重みベクトル $\boldsymbol{W} = \begin{bmatrix} 0.0 & 2.5 & 8.7 \\ 3.0 & 5.0 & 2.3 \\ 2.0 & -1.5 & -5.0 \end{bmatrix}$ になり、識別関数の最大値は $g[2]$ で、つまり、識別結果は雨という誤識別結果になる。なお、ρ の識別値を 0 から 2 まで、刻み幅 0.1 で実験した結果、全ては誤識別であり、ρ の幅とはほぼ無関係であることがわかった。

2.3　考察

各学習パターンを観察した結果、晴れの学習パターンはほとんど第一象限に所存して、評価パターンは第 3 象限にあるため、学習パターンは不適切であると想定する。一つの改良法として、評価パターンの近くで新たに学習パターンを観測し、新学習パターンを含め重みベクトルの修正を行う。パターンをそれぞれ学習パターンに加わり、識別結果を表 3 で表す。

表 3　追加パターンおよび識別結果

パターン名	x 座標	y 座標	天気	修正された重み	識別結果
New1	- 0.9	-0.5	晴れ	$\begin{bmatrix} 2.0 & -3.6 & 10.8 \\ 2.0 & 7.6 & -5.3 \\ -6.0 & -5.0 & -2.5 \end{bmatrix}$	晴れ
New2	- 0.7	-0.3	晴れ	$\begin{bmatrix} 1.0 & -2.6 & 9.5 \\ 1.0 & 3.0 & -2.4 \\ -4.0 & -1.4 & -4.1 \end{bmatrix}$	曇り

よって、パターン New1 が正しく識別できるようになったが、New2 が誤識別になった結果がわかった。

ゆえに、応用を通して、パーセプトロン法がクラスター分界線付近のパターンの識別力が低いという特徴がわかった。なお、観測データの選び方は識別結果に大きな影響を与え得ることがわかった。実際に応用する際、多層パーセプトロン、いわゆるニューラルネットワークによる識別する方が頼りになると考えられる。

参考文献

[1] 石井健一郎, 上田修功, 村瀬洋, 等. わかりやすいパターン認識 [M]. Ohmsha, 1998.