パターン認識　レポート

（再代入法と分割法による誤識別率の推定）

1. アルゴリズムとシミュレーション内容の説明

(1) 再代入法とは
(2) 分割法とは
(3) 各クラス 50 個のサンプルをどのように用いたか

(1) 再代入法とは

再代入法（Resubstitution 法）は、与えられた、利用できるすべてのサンプルを訓練サンプルとして用い、また、再びテストサンプルとしても用いる手法である。
この手法の長所は、「簡単さ」であり、短所は、「真の誤識別率よりも低い方へ偏った誤識別率が得られる」ことである。低い方へ誤識別率が偏るのは、一度学習したサンプルをテストに用いるために、本来の実力以上の良い性能となっているからであると、直感的に解説することができる。つまり、「真の誤識別率よりも低い誤識別率（負の偏り）」となる原因は、「訓練サンプルとテストサンプルの独立性がない」ことである。この独立性を保つ一手法として分割法（Holdout 法）が考えられている。

＜プログラムのアルゴリズム＞
①変数や配列の宣言と初期化
②プログラムにデータを渡す
再代入法では、扱う訓練データ、テストデータは同一なので、各ファイル（クラス）から配列に input したデータをそのまま使用している。
③50 個の訓練サンプルを用いて平均ベクトルの推定（式 2.39）
④50 個の訓練サンプルを用いて共分散行列の推定(式 2.40)
⑤共分散行列の逆行列と行列式の計算
⑥50 個のテストサンプルと各クラスとの距離を計算する（式 2.43　d1,d2,d3）
但し、事前確率は等確率なので、d1,d2,d3 に共通。
また、log 計算処理において、事前確率を 0.333 とすると、計算結果に誤差が生じてしまうため、

$$\frac{1}{2}\log|\Sigma| + \log 3$$

と式変形することによって、テストサンプルと各クラスとの距離を算出した
この処理を、for 文を用いることにより、50 個のテストサンプル全てにおいて、各クラスとの距離を求めている

⑦最も距離が近い（d の値が小さい）クラスへテストサンプルを識別する（式 2.42）
⑧識別されたテストサンプルが、正しいクラスへ識別されてるかをチェックする
　配列 judge[3]を使用し、（配列の添え字の 0 ~ 2 はクラス 1 ~ 3 に対応している）最小の距離を持つクラスに対応している配列の要素をインクリメントすることで、各クラスに判定された個数をカウントしている
⑨以上の①～⑧までの動作を、クラス setosa, versicolor, virginica における全テストサンプル 150 個に対して識別を行い、誤識別されたテストサンプル数を数える
⑩誤識別率を求める（全テストサンプルの中で何個間違ったかの割合を求める）

(2) 分割法とは

分割法（Holdout 法）とは、利用できるサンプルを半分に分け、一方を訓練サンプル、片方をテストサンプルとする手法である。
この手法の長所は、訓練サンプルとテストサンプルの分離による「訓練サンプルとテストサンプルの独立性がある」ことであり、短所は「真の誤識別率よりも大きい方へ偏った推定値が得られる」ことと、「サンプルの利用効率が悪い」ことである。「真の誤識別率よりも高い誤識別率（正の偏り）」となる原因は、元々、パターン認識問題では利用できる少ないサンプル数を分割するため、「訓練サンプル数がより少なくなる」ことである。サンプル数を増加させることは一般に困難であるため、サンプルの利用効率を高める leave-one-out 法が提案されている。

＜プログラムのアルゴリズム＞
①変数や配列の宣言と初期化
②プログラムにデータを渡す
分割法では、扱う訓練データ、テストデータは独立しているので、各ファイル（クラス）から配列に input したデータを、訓練サンプル用配列、テストサンプル用配列に、①～④の分割方法で格納している
③25 個の訓練サンプルを用いて平均ベクトルの推定（式 2.39）
④25 個の訓練サンプルを用いて共分散行列の推定(式 2.40）
⑤共分散行列の逆行列と行列式の計算
⑥25 個のテストサンプルと各クラスとの距離を計算する（式 2.43　d1,d2,d3）
但し、事前確率は等確率なので、d1,d2,d3 に共通。

また、log 計算処理において、事前確率を 0.333 とすると、計算結果に誤差が生じてしまうため、

$$\frac{1}{2}\log|\Sigma| + \log 3$$

と式変形することによって、テストサンプルと各クラスとの距離を算出した
この処理を、for 文を用いることにより、25 個のテストサンプル全てにおいて、各クラスとの距離を求めている
⑦最も距離が近い（d の値が小さい）クラスへテストサンプルを識別する（式 2.42）
⑧識別されたテストサンプルが、正しいクラスへ識別されてるかをチェックする
配列 judge[3]を使用し、（配列の添え字の 0 ~ 2 はクラス 1 ~ 3 に対応している）最小の距離を持つクラスに対応している配列の要素をインクリメントすることで、各クラスに判定された個数をカウントしている
⑨以上の①～⑧までの動作を、クラス setosa, versicolor, virginica における全テストサンプル 75 個に対して識別を行い、誤識別されたテストサンプル数を数える
⑩誤識別率を求める（全テストサンプルの中で何個間違ったかの割合を求める）

(3) 各クラス 50 個のサンプルをどのように用いたか

・再代入法では、各クラス（setosa, versicolor, virginica）の計 150 個を訓練サンプルとし、また、同様に計 150 個をテストサンプルとした。

・分割法では、各クラス 50 個のサンプルのうち、
①前半 25 個を訓練サンプル、後半 25 個をテストサンプルとした
②後半 25 個を訓練サンプル、前半 25 個をテストサンプルとした
③偶数番を訓練サンプル、奇数番をテストサンプルとした
④奇数番を訓練サンプル、偶数番をテストサンプルとした

分割法では、上記の①～④の分割の種類ごとにプログラムを変更し、それぞれ Holdout_1.c～Holdout_4.c としている。

2. 結果とその説明

(1) 再代入法、分割法（4 回それぞれ）の誤識別率（%）
(2) 再代入法では、各クラス、どのサンプルが誤識別されたのか
(3) 分割法（4 回それぞれ）各クラス、どのサンプルが誤識別されたのか
　また、①、②、③、④それぞれで誤識別率を推定し、それらの平均値を求める

次ページ以降に、
・再代入法（Resubstitution.c）、
・分割法①～④（Holdout_1.c ~Holdout_4.c）
の計 5 つのプログラムの実行結果を掲載する。

Resubstitution_result.txt ［ 再代入法による実行結果 ］

/* 実行結果 */

[tino@localhost ~]$ gcc Resubstitution.c -o Resubstitution.out -lm
[tino@localhost ~]$ ./Resubstitution.out

----setosa----
setosa と判定された数:50
versic と判定された数:0
virgin と判定された数:0

--versicolor--
setosa と判定された数:0
versic と判定された数:47
virgin と判定された数:3

virginica と誤識別されたテストサンプル番号及びベクトル成分
versicolor_number 21： 5.9 3.2 4.8 1.8
versicolor_number 34： 6.0 2.7 5.1 1.6
versicolor_number 40： 5.5 2.5 5.0 1.3

--virginica--
setosa と判定された数:0
versic と判定された数:1
virgin と判定された数:49

versicolor と誤識別されたテストサンプル番号及びベクトル成分
virginica_number 34： 6.3 2.8 5.1 1.5

全テストサンプル 150 個中 4 個 誤識別
誤識別率:2.667%
[tino@localhost ~]$

Holdout_1_result.txt　[分割法①による実行結果]

/* 実行結果 */

[tino@localhost ~]$ gcc Holdout_1.c -o Holdout_1.out -lm
[tino@localhost ~]$ ./Holdout_1.out

----setosa----
setosa と判定された数:25
versic と判定された数:0
virgin と判定された数:0

--versicolor--
setosa と判定された数:0
versic と判定された数:23
virgin と判定された数:2

virginica と誤識別されたテストサンプル番号及びベクトル成分
versicolor_number 9：　6.0　2.7　5.1　1.6
versicolor_number 15：　5.5　2.5　5.0　1.3

--virginica--
setosa と判定された数:0
versic と判定された数:1
virgin と判定された数:24

versicolor と誤識別されたテストサンプル番号及びベクトル成分
virginica_number 9：　6.3　2.8　5.1　1.5

全テストサンプル 75 個中 3 個 誤識別
誤識別率:4.000%
[tino@localhost ~]$

Holdout_2_result.txt　[分割法②による実行結果]

/* 実行結果 */

```
[tino@localhost ~]$ gcc Holdout_2.c -o Holdout_2.out -lm
[tino@localhost ~]$ ./Holdout_2.out

----setosa----
setosa と判定された数:25
versic と判定された数:0
virgin と判定された数:0

--versicolor--
setosa と判定された数:0
versic と判定された数:23
virgin と判定された数:2

virginica と誤識別されたテストサンプル番号及びベクトル成分
versicolor_number 19：　6.2　　2.2　　4.5　　1.5
versicolor_number 21：　5.9　　3.2　　4.8　　1.8


--virginica--
setosa と判定された数:0
versic と判定された数:0
virgin と判定された数:25

versicolor と誤識別されたテストサンプル番号及びベクトル成分
no data


全テストサンプル 75 個中 2 個 誤識別
誤識別率:2.667%
[tino@localhost ~]$
```

Holdout_3_result.txt　[分割法③による実行結果]

/* 実行結果 */

```
[tino@localhost ~]$ gcc Holdout_3.c -o Holdout_3.out -lm
[tino@localhost ~]$ ./Holdout_3.out

----setosa----
setosa と判定された数:25
versic と判定された数:0
virgin と判定された数:0

--versicolor--
setosa と判定された数:0
versic と判定された数:23
virgin と判定された数:2

virginica と誤識別されたテストサンプル番号及びベクトル成分
versicolor_number 10：　6.2　　2.2　　4.5　　1.5
versicolor_number 11：　5.9　　3.2　　4.8　　1.8


--virginica--
setosa と判定された数:0
versic と判定された数:0
virgin と判定された数:25

versicolor と誤識別されたテストサンプル番号及びベクトル成分
no data


全テストサンプル 75 個中 2 個 誤識別
誤識別率:2.667%
[tino@localhost ~]$
```

Holdout_4_result.txt　[分割法④による実行結果]

/* 実行結果 */

```
[tino@localhost ~]$ gcc Holdout_4.c -o Holdout_4.out -lm
[tino@localhost ~]$ ./Holdout_4.out

----setosa----
setosa と判定された数:25
versic と判定された数:0
virgin と判定された数:0

--versicolor--
setosa と判定された数:0
versic と判定された数:23
virgin と判定された数:2

virginica と誤識別されたテストサンプル番号及びベクトル成分
versicolor_number 17：　6.0　　2.7　　5.1　　1.6
versicolor_number 20：　5.5　　2.5　　5.0　　1.3


--virginica--
setosa と判定された数:0
versic と判定された数:2
virgin と判定された数:23

versicolor と誤識別されたテストサンプル番号及びベクトル成分
virginica_number 16：　7.9　　3.8　　6.4　　2.0
virginica_number 17：　6.3　　2.8　　5.1　　1.5


全テストサンプル 75 個中 4 個 誤識別
誤識別率:5.333%
[tino@localhost ~]$
```

(1) 再代入法における誤識別率は 2.667 (%)

分割法①における誤識別率は 4.000 (%)
分割法②における誤識別率は 2.667 (%)
分割法③における誤識別率は 2.667 (%)
分割法④における誤識別率は 5.333 (%)

(2) 再代入法で誤識別されたサンプル：

virginica と誤識別された「**クラス：versicolor**」のテストサンプル番号及びベクトル成分

versicolor_number 21：	5.9	3.2	4.8	1.8
versicolor_number 34：	6.0	2.7	5.1	1.6
versicolor_number 40：	5.5	2.5	5.0	1.3

versicolor と誤識別された「**クラス：virginica**」のテストサンプル番号及びベクトル成分

virginica_number 34：	6.3	2.8	5.1	1.5

(3) (a)分割法①で誤識別されたサンプル：

virginica と誤識別された「**クラス：versicolor**」のテストサンプル番号及びベクトル成分

versicolor_number 9：	6.0	2.7	5.1	1.6
versicolor_number 15：	5.5	2.5	5.0	1.3

versicolor と誤識別された「**クラス：virginica**」のテストサンプル番号及びベクトル成分

virginica_number 9：	6.3	2.8	5.1	1.5

※テストサンプルは、各クラスの後半の 25 個を使用しているので、実際のサンプル番号にするためには、求めたサンプル番号に「25 を足す」

(b)分割法②で誤識別されたサンプル：

virginica と誤識別された「**クラス：versicolor**」のテストサンプル番号
及びベクトル成分

versicolor_number 19：	6.2	2.2	4.5	1.5
versicolor_number 21：	5.9	3.2	4.8	1.8

(c)分割法③で誤識別されたサンプル：

virginica と誤識別された「**クラス：versicolor**」のテストサンプル番号
及びベクトル成分

versicolor_number 10：	6.2	2.2	4.5	1.5
versicolor_number 11：	5.9	3.2	4.8	1.8

※テストサンプルは、各クラスの奇数番の 25 個を使用しているので、実際のサンプル番号にするためには、求めたサンプル番号を「2 倍して 1 を引く」

(d)分割法④で誤識別されたサンプル：

virginica と誤識別された「**クラス：versicolor**」のテストサンプル番号
及びベクトル成分

versicolor_number 17：	6.0	2.7	5.1	1.6
versicolor_number 20：	5.5	2.5	5.0	1.3

versicolor と誤識別された「**クラス：virginica**」のテストサンプル番号
及びベクトル成分

virginica_number 16：	7.9	3.8	6.4	2.0
virginica_number 17：	6.3	2.8	5.1	1.5

※テストサンプルは、各クラスの偶数番の 25 個を使用しているので、実際のサンプル番号にするためには、求めたサンプル番号を「2 倍する」

また、分割法①、②、③、④のそれぞれの誤識別率の平均値を求めると
（4.000 + 2.667 + 2.667 + 5.333）/ 4 = 3.66675

よって、4 種類の分割方法における、分割法の**平均値**は「**3.66675**」となる。

この分割法の誤識別率の平均値「3.66675」は、再代入法の誤識別率「2.667」よりも大きいことが確認できる。

3. 考察

(1) 分割法の 4 つの誤識別率のバラツキについて
(2) 再代入法と分割法との比較について

(1) 分割法の 4 つの誤識別率のバラツキについて

分割法の 4 つ誤識別率は、各 iris データの奇数番目を訓練サンプルとした場合、最も誤識別率が高くなり（5.333 (%)）、各 iris データの後半 25 個を訓練サンプルとした場合、または偶数番目を訓練サンプルとした場合に最も誤識別率が低くなった（2.667 (%)）。この 2 つの分割方法による誤識別率「2.667%」は、真の誤識別率よりも低い方へ推定値が偏る「再代入法」における誤識別率と一致したので、iris データを分割する場合は、「後半 25 個を訓練サンプルとする」または、「偶数番を訓練サンプルとする」と、誤識別率がより低くなり、精度の高い識別を行うことができると考察できる。

また、4 つの分割方法において、いずれも“クラス setosa ”のデータは、“クラス versicolor”、“クラス virginica”に誤識別されていなかったので、setosa は、他の 2 つのクラスから離れた位置に分布しているクラスであると考察できる。
逆に、4 つの分割方法において、“クラス versicolor”、“クラス virginica”のデータは、互いのクラスに誤識別されていたことから、verginica, versicolor は似通ったデータ分布であると考察できる。

さらに、誤識別されたデータのベクトル成分を見てみると、virginica と誤識別された「クラス : versicolor」のテストサンプルは、いずれも同じクラス内の他のテストサンプルに比べ、ベクト成分 x3 の値が比較的に大きいことが分かる。
また、versicolor と誤識別された「クラス : virginica」のテストサンプルは、いずれも同じクラス内の他のテストサンプルに比べ、ベクトル成分 x3 の値が比較的に小さいことが分かる。
このことから、クラス versicolor のデータよりもクラス virginica のデータ（特にベクトル成分 x3）の方が大きいのではないかと考察した。

(2) 再代入法と分割法との比較について

一般的に、再代入法は、真の誤識別率率よりも低い方へ偏った推定値となり、分割法は、真の誤識別率よりも高い方へ偏った推定値が得られる。
今回、分割法では、4 種類の分割方法で誤識別率の推定を行ったが、誤識別率はいずれも再代入法で得られた誤識別率を下回ることはなかった。よって、この誤識別率推定結果は妥当であると考察することができる。
また、この結果から、再代入法で得られた誤識別率 2.667%と、分割法で得られた誤識別率の平均値である 3.66675%の間に、真の誤識別率が存在するのではないかと考察した。

4. 付録

次ページ以降に、

・再代入法のプログラム「Resubstitution.c」
・分割法①のプログラム「Holdout_1.c」
・分割法②のプログラム「Holdout_2.c」
・分割法③のプログラム「Holdout_3.c」
・分割法④のプログラム「Holdout_4.c」

及び、各実行結果のキャプチャ画面を掲載する。