

R と判別分析

1. 判別分析とは

私たち人間は毎日五感を通じて入力される膨大なデータを処理している。その中で最も多いのは、識別(discrimination)、分類(classification)、認識(recognition)に関する処理である。例えば、新聞や本などを読むときには、視覚を通じて入力されたデータと学習したデータとの照合を行い、その文字の読み方、文字・単語の意味などを識別・認識する。識別に関する能力は人間のみならず、他の動物も持っている。

このような識別・認識に関することを機械的に実現する研究分野がパターン認識(pattern recognition)である。パターン認識の典型的な例としては、郵便番号による手紙の自動分類や指紋・顔照合によるセキュリティ管理などがあげられる。

パターン認識は、コンピュータに事前に入力・記憶させたデータと識別すべきデータとの一致度を何らかのモデルによって計算する。その中、最も古典的な手法は、判別分析(discriminant analysis)である。

判別分析は、個体(あるいは対象)がどのグループに属するかが明確である学習データを用いて判別モデルを構築し、そのモデルを用いて所属不明の個体(テスト用データ)がどのグループに帰属するかを判別する方法である。

古典的判別分析には、距離(あるいは類似度)による判別と判別関数による判別分析などの方法がある。

判別分析では、所属不明の個体が2つのグループのいずれに属するかを判別する問題を2群判別分析、3つ以上のグループのいずれかに属するかに関する判別問題を多群判別分析とよぶ。

判別問題における学習データは、次の表のようにどの個体がどのグループに属するかに必要な情報が必要である。グループに関する情報は、回帰分析の被説明変数に対応するが、回帰分析の場合は被説明変数が量的であるのに対して、判別分析のグループに関する変数は質的変数である。

表1 グループ G_1 、 G_2 に関する学習用データ

個体	変	数	グループ情報
1	$x_{11}^{(1)}$ $x_{12}^{(1)}$...	$x_{1n}^{(1)}$	1
2	$x_{21}^{(1)}$ $x_{22}^{(1)}$...	$x_{2n}^{(1)}$	1
⋮	⋮	⋮	⋮
m	$x_{m1}^{(1)}$ $x_{m2}^{(1)}$...	$x_{mn}^{(1)}$	1

1	$x_{11}^{(2)}$	$x_{12}^{(2)}$...	$x_{1n}^{(2)}$	2
2	$x_{21}^{(2)}$	$x_{22}^{(2)}$...	$x_{2n}^{(2)}$	2
\vdots	\vdots	\vdots		\vdots	\vdots
l	$x_{l1}^{(2)}$	$x_{l2}^{(2)}$...	$x_{ln}^{(2)}$	2

表 1 の中の $x_{ij}^{(k)}$ はグループ G_k に属する i 番目の個体の j 番目の変数である。グループ G_1 に属する個体は m 個、グループ G_2 に属する個体は l 個である。

判別分析の問題では、グループを識別するためのモデル(規則、あるいは関数など)を求めるためのデータセット(学習データ)とそのモデルを用いて判別、あるいは評価を行うデータセット(テストデータ)が必要である。

2. 距離による判別分析

ここでは、説明の便利のため 2 群判別分析を例とする。距離による判別分析では、まず 2 つのグループ G_1 と G_2 の中心 $\mu^{(1)}$ 、 $\mu^{(2)}$ を求める。中心として最も多く用いられているのはグループの平均ベクトルである。

グループ G_1 、 G_2 のそれぞれの平均ベクトルを $\mu^{(1)}$ 、 $\mu^{(2)}$ とする。

$$\begin{aligned}\mu^{(1)} &= (\mu_1^{(1)}, \mu_2^{(1)}, \dots, \mu_n^{(1)}) \\ \mu^{(2)} &= (\mu_1^{(2)}, \mu_2^{(2)}, \dots, \mu_n^{(2)})\end{aligned}$$

この $\mu_j^{(1)}$ 、 $\mu_j^{(2)}$ は次のように定義されている。

$$\begin{aligned}\mu_j^{(1)} &= \frac{1}{m} \{x_{1j}^{(1)} + x_{2j}^{(1)} + \dots + x_{mj}^{(1)}\} \\ \mu_j^{(2)} &= \frac{1}{l} \{x_{1j}^{(2)} + x_{2j}^{(2)} + \dots + x_{lj}^{(2)}\} \\ &(\text{ただし } j = 1 \sim n)\end{aligned}$$

次に所属不明の個体からグループ G_1 、 G_2 の中心までの距離を求め、所属不明の個体は距離の小さい方のグループに属すると判断する。

今、所属不明の個体 H があるとする。

$$H = (x_{h1}, x_{h2}, \dots, x_{hn})$$

個体 H からグループ G_1 の中心 $\mu^{(1)}$ までの距離を D_1 、グループ G_2 の中心 $\mu^{(2)}$ までの距離を D_2 とする。

距離による判別分析では、 $D_1 < D_2$ ならば個体 H はグループ G_1 に、 $D_1 > D_2$ ならば個体 H はグループ G_2 に属すると判断する。

判別すべき個体が2つのグループの中心から等距離にある特殊な場合は、判別不能である。

このような距離を用いた判別分析法は、グループ数が3以上の場合にも簡単に拡張することができる。またデータに関しては、どのような確率分布に従っているかのような条件を必要としないのが長所である。

距離の測度としては、Rにはユークリッド距離、市街距離、マハラノビス距離などの関数が用意されている。

マハラノビス(mahalanobis)距離は、多変量データ解析の書物に必ず登場する距離で、マハラノビス汎距離とも呼ぶ。マハラノビス距離は次のように定義されている。

$$D(X, \mu) = \{(X - \mu)^t S^{-1} (X - \mu)\}^{\frac{1}{2}}$$

式の中の X はデータセット、 μ はグループの中心、 S はそのグループの分散共分散行列、 S^{-1} は分散共分散行列の逆行列である。

Rにはマハラノビス距離を求める関数 **mahalanobis** がある。その書き方を次に示す。

mahalanobis(X , μ , S)

実例を用いて説明するため、Rの中に用意されている iris データを用いることにする。データ iris に関しては本誌の No.117(p.87)に説明されている。help(iris)コマンドを実行すると英文による説明を読むことができる。

ここでは問題を簡単にするため、iris データの第1行から第50行までの setosa という品種、第101行から第150行までの virginica という品種のデータを用いる。

データ iris の第1列から第4列まではそれぞれアヤメのがく片の長さ、幅、花弁の長さ、幅に関する計測データで、第5列(Species)は品種(グループ)を表す質的データである。

```
>data(iris)#バージョン 2.0 以後は必要ではない
>iris[c(1,51,101),]
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
51	7.0	3.2	4.7	1.4	versicolor
101	6.3	3.3	6.0	2.5	virginica

まず iris の中の setosa と virginica の品種別のデータセット seto、virgi を作成する。品種ごとにデータセットを作成するので、グループの属性を示す Species 列は必要がない。

```
> seto<-subset(iris[1:50,],select=-Species)
> virgi<-subset(iris[101:150,],select=-Species)
```

seto、virgi はそれぞれ 50 行のデータである。ここでは、各データセットの 1~45 行を学習データとし、残りの 5 行(46~50)をテストデータとする。

次に学習用データの平均ベクトルと分散共分散行列を求める。

```
> seto.m<-apply(seto[1:45,],2,mean)
```

```
> virgi.m<-apply(virgi[1:45,],2,mean)
>seto.v<-var(seto[1:45,])
>virgi.v<-var(virgi[1:45,])
```

次のコマンドを実行するとテストデータ seto[46:50]と両グループ(seto[1:45,]、virgi[1:45,])の平均とのマハラノビス距離が求められる。

```
>D1<-mahalanobis(seto[46:50,],seto.m,seto.v)
>D2<-mahalanobis(seto[46:50,],virgi.m,virgi.v)
```

求めた距離から seto[46:50,]が seto、virgiの両グループの中のどのグループに属するかを確認するため、両グループまでの距離を次に示す。

```
> cbind(D1,D2)
      D1      D2
46 2.1752192 137.9376
47 2.8163645 173.8815
48 1.4346178 142.1425
49 1.2398930 182.5972
50 0.4700029 160.2070
```

D1 は seto[46:50,]と seto の中心、D2 は seto[46:50,]と virgi の中心との距離である。各行のD1列の値がD2列の値より小さいので seto[46:50,]は seto のグループに属すると判断する。

同様な方法で virgi[46:50,]と両グループの中心までのマハラノビス距離を求め、どのグループに属するかを判別することができる。

説明の便利のため、ここでは学習用のデータとテスト用のデータを恣意的に前の45行と後の5行に分けているが、現実問題では、もっと説得力のある方法を取るべきである。

マハラノビス距離の定義からわかるように、マハラノビス距離を求めるためには、学習データの分散共分散行列の逆行列を求める必要がある。しかし、すべてのデータの分散共分散行列の逆行列が求められるとは限らない。

3. 判別関数による判別分析

判別関数による判別分析は、線形関数と非線形関数による判別分析に分けられる。

(1) 線形判別分析

線形判別分析(Linear Discriminant Analysis)は、グループ分けの境界が直線、あるいは超直面であり、次のような線形関数を用いてグループの所属の判別を行う方法である。

$$\text{判別関数} = a_0 + a_1x_1 + a_2x_2 + \dots + a_nx_n$$

判別関数は、グループ間の分散とグループ内の分散の比を最大化することで求めることがで

きる。

Rのパッケージ MASSの中にこのアプローチによる判別分析の関数オブジェクト ldaがある。

```
>library(MASS)
```

関数 ldaの詳細については、help(lda)で読むことができる。次に ldaの最も簡潔な書式を示す。

```
lda(formula, data)
```

formulaには、回帰分析の場合と同様、「グループの識別変数~変数」のように記述する。ここでも Rの中の irisデータを用いて、ldaの使用法について説明する。

データセットの作成

前項では説明の便利のため、irisの2種類のデータを用いたがここでは3種類の irisを用いて3群判別分析を行う。

まずデータ irisから、学習データとテストデータを作成する。その作成方法はいろいろ考えられるが、ここでは奇数行と偶数行に二分することにする。

```
>even.n<-2*(1:75)-1  
>train.data<-iris[even.n,]  
>test.data<-iris[-even.n,]
```

上記のコマンドで150行の中から行番号が奇数である75行を学習データ(train.data)、残りの偶数行のデータをテストデータ(test.data)とするデータセットが作成される。

この段階で判別分析を行っても良いが、データ irisの中のグループを示すラベルが若干長いので、散布図を作成すると見苦しくなる。そこで、setosaをS、versicolorをC、virginicaをVで示すことにする。

```
>Iris.lab<-factor(c(rep("S",25),rep("C",25),rep("V",25)))  
>train.data[,5]<-Iris.lab  
> train.data[c(1,26,51),]
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	S
51	7.0	3.2	4.7	1.4	C
101	6.3	3.3	6.0	2.5	V

判別関数を求める

学習データセットを次ぎのように用いると判別分析に必要な統計量が求められる。ただし、次の書き式は学習データセットの中の変数をすべて用いた場合の書き方である。学習データセットの中の変数の1部分を用いる際には、 $y \sim x_1 + x_3$ のように用いる変数を記号「+」でつ

なく。

```
>(Z<- lda(Species~ .,data=train.data))
Call:
lda(Species ~ ., data = train.data)

Prior probabilities of groups:
      C      S      V
0.3333333 0.3333333 0.3333333

Group means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
C           5.992         2.776         4.308         1.352
S           5.024         3.480         1.456         0.228
V           6.504         2.936         5.564         2.076

Coefficients of linear discriminants:
              LD1      LD2
Sepal.Length -0.5917846 -0.1971830
Sepal.Width  -1.8415262  2.2903417
Petal.Length  1.6530521 -0.7406709
Petal.Width   3.5634683  2.6365924

Proportion of trace:
  LD1  LD2
0.9913 0.0087
```

返された線形判別係数(Coefficients of linear discriminants)を用いて判別関数を構築する。判別関数は、用いる変数と判別係数との線形結合である。

この問題では2組の判別係数(LD1、LD2)が返されている。この判別係数を用いた第1判別関数を次に示す。

$$f_{LD1} = -0.592x_1 - 1.842x_2 + 1.653x_3 + 3.564x_4 - c$$

判別関数式の中の x_1, x_2, x_3, x_4 はそれぞれ iris データの変数 Sepal.Length、Sepal.Width、Petal.Length、Petal.Width を示す。

定数項 c はグループの平均と判別係数を次ぎのように用いて求めることができる。第1列の値は第1判別関数の定数項で、第2列は第2判別関数の定数項である。

```
> apply(Z$means%*%Z$scaling,2,mean)
      LD1      LD2
1.486146 6.282412
```

上記の判別関数を用いた判別得点はコマンド `predict(Z)$x` で返される。

判別関数で得られた値を判別得点(discriminant score)とよぶ。各判別関数が全体のグループ間分散

をどのくらい説明できるかはグループ間分散の比率から読み取れる。

学習データにおける判別結果

判別関数を用いて、学習データについて判別を行った結果は関数 `predict` を用いて返す。関数 `predict` は次の値を返す。

```
predict()$class  
predict()$posterior  
predict()$x
```

`$class` は各個体が判別されたグループのラベルで、`$posterior` は各個体がどのグループに判別されているかに関する事後確率 (0 ~ 1)、`$x` は各個体の判別関数得点である。

学習データにおける判別結果は次のような表で確認することができる。

```
> table(train.data[,5],predict(Z)$class)
```

```
  C  S  V  
C 24  0  1  
S  0 25  0  
V  1  0 24
```

Cグループの1つがVグループに、Vグループの1つがCグループに誤判別され、誤判別率は $2/75 = 0.0267$ (2.67%) である。どの個体が誤判別されているかは、次のようなコマンドで追跡することができる。

```
> data.frame(train.data[,5],predict(Z)$class)
```

判別関数得点をグラフに示すこともできる。次のコマンドで第1判別関数得点のグループごとのヒストグラム(分布)が作成される。

```
> plot(Z,dimen=1)
```

作成されたヒストグラムから分かるように、S(setosa)はC(versicolor)、V(virginica)と重ならないが、C(versicolor)の右辺とV(virginica)の左辺は若干重なる。重なる領域が多いほどお互いに間違っ判別される確率(誤判別率)が高い。

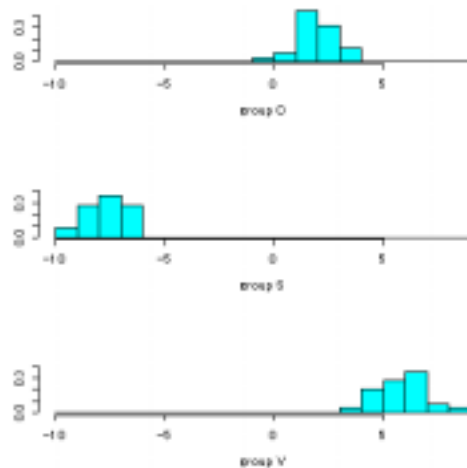


図1 学習データの第1判別関数得点の分布

引数 `dimen=2` を用いると、図2のような、横軸を第1判別関数、縦軸を第2判別関数とした散布図が作成される。散布図からグループ間の分類状況がマクロ的に把握できる。判別関数が3つ以上の場合は、`dimen` の値を3以上にすると対散布図が作成される。

```
>plot(Z,dimen=2)
```

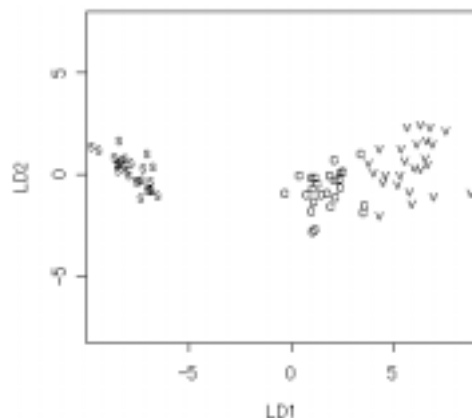


図2 学習データの判別関数得点の散布図

判別関数による判別

学習データで求めた判別関数を用いてテストデータについての判別分析は次のように関数 `predict` を用いる。

```
>Y<-predict(Z,test.data)
```

どれくらい正しく判別されているかは、テストに用いたデータのグループ情報と判別結果のグループ情報のクロス表を作成することで確認できる。

```
>table(test.data[,5],Y$class)
```

```
  C  S  V
C 24  0  1
S  0 25  0
V  2  0 23
```

上記の結果から分かるように、テストデータの中の S(setosa)はすべて正しく判別され、C(versicolor)は1つが V(virginica)と誤判別され、Vは2つが C と誤判別されている。誤判別されているのは3個で全体の中で占める割合(誤判別率)は $3/75 = 0.04(4\%)$ 、正判別率は $1-0.04=0.96(96\%)$ である。

テストデータの判別得点の散布図を用いて、判別状況を視覚的に考察することもできる。

```
>plot(Y$x,type="n")
>text(Y$x,labels=as.character(Iris.lab))
```

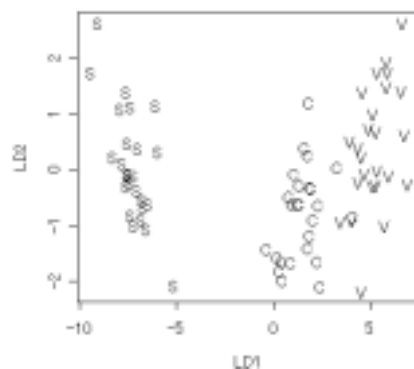


図3 テストデータの判別関数値の散布図

このアプローチによる線形判別分析は、各グループの母分散が等しいであるとの仮定に基づいている。各グループの母分散が異なる場合は、マハラノビス距離による判別のような母分散に制約がない方法を用いるべきである。

(2) 非線形判別分析

古典的な非線形判別分析では、2次式を含む非線形関数を用いる方法がある。Rのパッケージ MASS 中には2次式による判別分析の関数 qda が用意されている。

関数 qda の使用方法は、lda とほとんど同じである。

一般化線形モデルで紹介した、目的変数が質的変数であるロジスティック回帰も非線形判別分析の一つの方法である。

判別関数のアプローチによる判別分析は、属性を示す変数の数が多くなると判別関数の構築が難しくなる。

4. 交差確認

判別分析やパターン認識の分野では、データセットから学習用のデータとテスト用のデータ

に分けてモデルの構築と評価をするのに交差確認(cross validation : 交差検証、交差妥当化とも訳されている)という方法が広く知られている。

交差確認では、データセット全体を n 部分(サブデータセット)に均等に分割し、そのうちの1つをテスト用のデータとして残し、それ以外の $n-1$ つを学習用のデータとして用いる。

データセットを n 部分に分割したときを n 重交差確認(n -fold cross validation)法と呼ぶ。 n 重交差確認法では、一つのデータセットに対し、 n 回のモデルの構築とテスト(確認、検証)を行い、その n 回のテスト結果を全体の評価に用いる。

関数 `lda`、`qda` では、データセットから1つの個体を除いて学習を行い、学習データに用いていない1つの個体で判別モデルの評価を行う作業を、すべての個体に対して繰り返す交差確認(leave-one-out cross-validation)の引数 `CV` が用意されている。これは、 n 重交差確認の n が個体の数に等しい特殊なケースである。

デフォルトでは `CV=FALSE` になっている。引数 `CV=T` にすると1つを除いた交差確認による結果が返される。次に `iris` データを用いた1つを除いた交差確認の結果を示す。

```
> iris.CV<-lda(Species~.,data=iris,CV=T)
>(lda.tab<- table(iris[,5],iris.CV$class))
      setosa versicolor virginica
setosa    50         0         0
versicolor 0         48         2
virginica  0          1        49
```

判別率と誤判別率は次のコマンドで求めることができる。

```
> sum(lda.tab[row(lda.tab)==col(lda.tab)])/sum(lda.tab)
[1] 0.98
> sum(lda.tab[row(lda.tab)!=col(lda.tab)])/sum(lda.tab)
[1] 0.02
```

個体数が十分大きくない古典的な判別分析では、1つの個体を除いた交差確認法が多く用いられているが、ニューラルネットワーク、決定木、サポートベクトルマシンのようなパターン認識の方法では、一般の n 重交差確認法が多く用いられている。

n をいくつにするべきであるかは、用いるデータのサイズに依存するため明確な基準がなく、 $n=2, 3, 4, 5, 10$ が多く用いられているのが現状である。

残念ながら、判別関数 `lda`、`qda` には leave-one-out cross-validation 以外の交差確認の機能が用意されていない。