非線形判別分析

1196508c　伊東　慧

1．非線形判別分析とは

- 線形判別関数以外による判別分析
- 今回用いる方法
  - 非線形関数による判別分析
  - 距離による判別分析
  - 多数決によるｋ最近傍法
  - ベイズ判別法

２．判別関数による判別分析

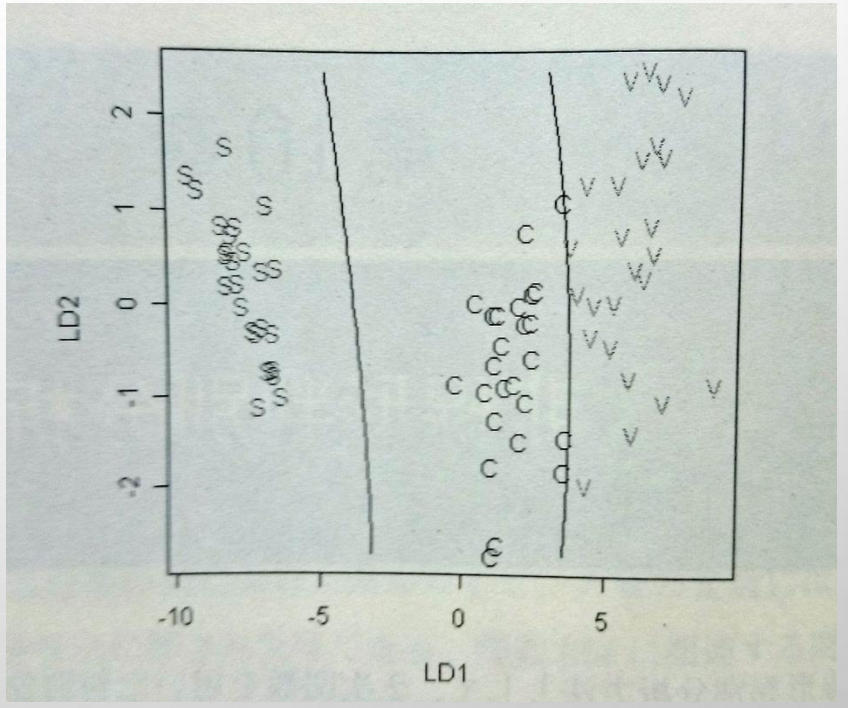
- 二次式を含む初等関数が多用されている
- 前章「線形判別分析」同様”iris”の学習データ、テスト用データを用いる
- 下準備（前章スライド4枚目と同様）

```
> iris.lab<- c(rep("S",50),rep("C",50),rep("V",50))
> iris1<-data.frame(iris[,1:4],Species=iris.lab)
> even.n<-2*(1:75)-1
> iris.train<-iris1[even.n,]
> iris.test<-iris1[-even.n,]
>
```

2. 判別関数による判別分析

- 前章「線形判別分析」で用いた関数”lda”と同じ書式の”qda”を用いる

```
> library(MASS)
> Z.qda<-qda(Species~.,iris.train)
> table(iris.train[,5],predict(Z.qda)$class)

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



判別得点の散布図と2時判別曲線の図
（コマンドライン過多により実行例は省略）

2. 判別関数による判別分析

- 求めた判別関数に基づいた、テスト用のデータの判別結果を求める

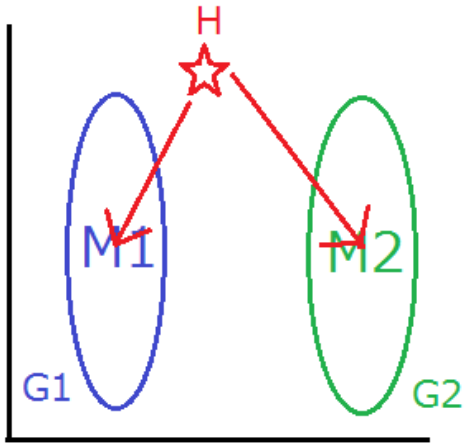
```
> Y.qda<-predict(Z.qda,iris.test[,-5])
> table(iris.test[,5],Y.qda$class)

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

- 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

  →目的変数が2値のロジスティック回帰分析（8章参考）を用いた2群判別分析が有効

3. 距離による判別分析

- ２郡判別分析を例とした説明
  1. 学習データからグループG_1，G_2 の中心M_1，M_2を求める
  2. 未知の個体Ｈとグループとの中心の距離を測る
  3. より距離が近いグループに属すると判断



- グループ数が３以上の場合も簡単に拡張できる
- データに関しどのような確率分布に従っているかという条件不要

３．距離による判別分析

- データの分散の情報を用いたマハラビノス距離が多く用いられる
  - 長所･･･理解しやすく、距離は自由に定義でき、ロバスト（外乱に強い）である
  - 短所･･･距離の計算に必要な分散共分散行列の逆行列がデータによって求められない

- マハラノビス距離を求める関数”mahalanobis”

```
> mahalanobis(x, center, cov, ...)
```

  - ｘ→データ　ｃｅｎｔｅｒ→中心ベクトル　ｃｏｖ→分散共分散行列
- 次スライドから再び“iris.train”　”iris.test”を用い実践

3. 距離による判別分析

- 学習データ“iris.train”の品種別の中心ベクトル、分散共分散行列を求める

```
> seto.mean<-apply(iris.train[1:25,-5],2,mean)
> seto.var<-var(iris.train[1:25,-5])
> vers.mean<-apply(iris.train[26:50,-5],2,mean)
> vers.var<-var(iris.train[26:50,-5])
> virg.mean<-apply(iris.train[51:75,-5],2,mean)
> virg.var<-var(iris.train[51:75,-5])
```

- 今回は中心ベクトルとして平均ベクトルを用いている
- データよっては中央ベクトルを用いたほうがよい場合も

3. 距離による判別分析

- 学習データ“iris.train”の各個体と各グループの中心との距離を求める

```
> D1<-mahalanobis(iris.train[,-5],seto.mean,seto.var)
> D2<-mahalanobis(iris.train[,-5],vers.mean,vers.var)
> D3<-mahalanobis(iris.train[,-5],virg.mean,virg.var)
```

- 各個体と３つのグループ中心との距離を示す

```
> round(cbind(D1,D2,D3),0)
      D1  D2  D3
1      0 118 173
3      1  98 148
5      1 123 174
7      3  98 141
           :
```

- 各個体は距離の値が小さいグループに属する

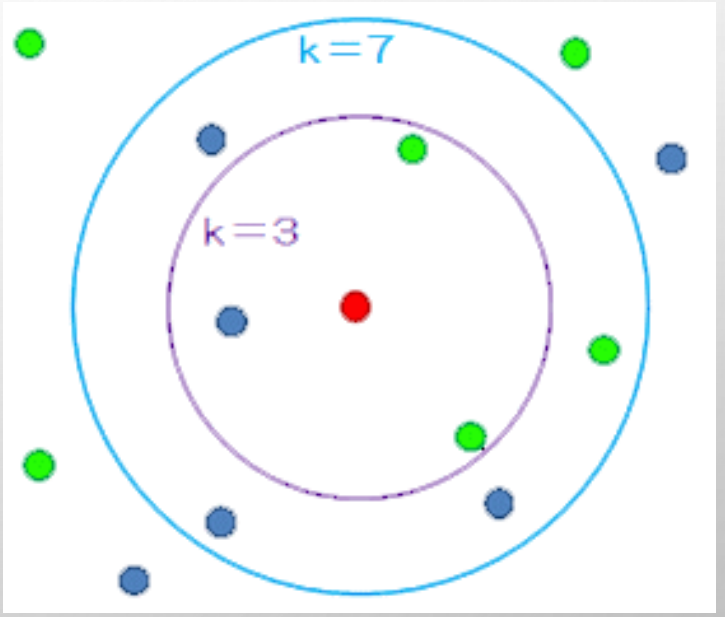
3. 距離による判別分析

- 学習データの中心ベクトルと分散共分散行列を用いたテスト用データの判別結果を示す

```
> D1.t<-mahalanobis(iris.test[,-5],seto.mean,seto.var)
> D2.t<-mahalanobis(iris.test[,-5],vers.mean,vers.var)
> D3.t<-mahalanobis(iris.test[,-5],virg.mean,virg.var)
> round(cbind(D1.t,D2.t,D3.t),0)
    D1.t D2.t D3.t
2      4   87  143
4      2   87  129
6      6  120  177
           :
```

4. 多数決による判別分析

- 伝統的なパターン分類アルゴリズム「k 最近傍法」
- 判別すべき個体の周辺の最も近いものをｋ個見つけ、その多数決によりグループを確定
- 例：右の図において●は
  - ｋ＝３のとき●のグループに属する
  - ｋ＝７のとき●のグループに属する
- ｋは自由であり、データに依存するため明確な基準なし



4. 多数決による判別分析

- パッケージ”class”にあるｋ最近傍法の関数”knn”

```
> knn(train,test,cl,k=1,...)
```

  - train→学習用データ　test→テスト用データ
  - cl→学習用データのグループの属性データ　ｋ＝（任意の数）→近傍の個体の数

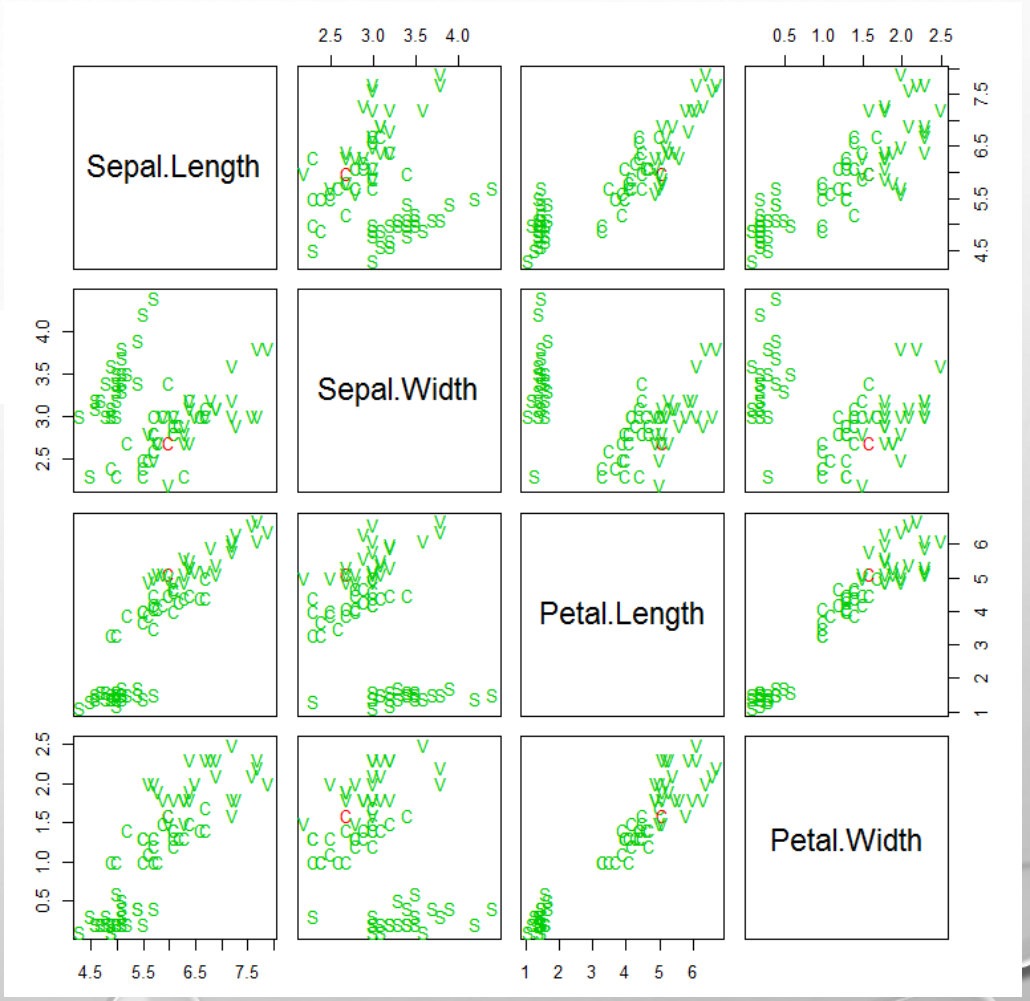
- ｋ＝５として“iris.train”　”iris.test”データにｋ最近傍法を用いる

```
> library(class)
> iris.knn<-knn(iris.train[,-5],iris.test[,-5],iris.train[,5],k=5)
> (table(iris.knn,iris.test[,5]))

iris.knn  C  S  V
       C 24  0  0
       S  0 25  0
       V  1  0 25
```

4. 多数決による判別分析

- データの変数が多くない場合には対散布図に誤判別された個体を異なる色で表示可能

```
> pairs(iris.test[,1:4],pch=as.character(iris.test[,5]),
+ col = c(3,2)[(iris.test$Species != iris.knn)+1])
```



4. 多数決による判別分析

- 用いたデータセットの中から１つを取り除き学習・テストを繰り返す交差確認法
  それを用いる関数”knn.cv”

```
> iris.cv<-knn.cv(iris[,-5],iris[,5],k=5)
> table(iris[,5],iris.cv)
            iris.cv
             setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         47         3
  virginica       0          2        48
```

- データ量が少なくテストデータの選び方によって推定制度の大きな誤差が
  生じうるときに有効

5. ベイズ判別法による判別分析

- ある個体ｘがあるグループｙに属する確率を求める
- 与えられた学習用のデータからその確率が最大となるモデルを求め、
  それらを基に所属不明の個体についてそのグループの所属を判別

- 個体の特徴が質的データである場合→相対頻度を確率の推測値にする
- 個体の特徴が量的データである場合→質的データに離散化する
  or　ある確率分布に属するという仮定を行う必要
  - 現実では後者のほうが多く用いられる ex.）正規分布、対数正規分布、ガンマ分布

5. ベイズ判別法による判別分析

- パッケージe1071やklaRなどに含まれるナイーブベイズ分類器の関数
- 分類器とは学習によって分類を行うシステム
- パッケージe1071の関数”naiveBayes”
  - 標準的なアルゴリズムを用いデータが独立で、正規分布に従うと仮定

```
> NaiveBayes(formula, data,...)
> NaiveBayes(x, grouping, prior, usekernel=FALSE)
```

- 因数”usekernel”　確率密度をカーネル法（第15章参照）で推定するか否かを設定
  データによっては結果に大きな影響を与える

5. ベイズ判別法による判別分析

- ガラス破片データ”glass”を用いる
- カーネル法を用いない

```
> install.packages("klaR"); library(klaR)
> install.packages("mlbench"); library(mlbench)
> data(Glass); G<-Glass[,c(1:5,10)]
> m1<-NaiveBayes(Type~.,data=G)
> m1.p<-predict(m1)
> tem1<-table(G$Type, m1.p$class)
> 1-sum(diag(tem1))/sum(tem1)
[1] 0.4766355
```

- カーネル法を用いる

```
> m2<-NaiveBayes(Type~.,data=G,usekernel=TRUE)
> m2.p<-predict(m2)
> tem2<-table(G$Type, m2.p$class)
> 1-sum(diag(tem2))/sum(tem2)
[1] 0.2102804
```

- 今回はカーネル法を用いることにより誤判別率が約27パーセント下がっている