判別分析
Native vs. Non-native

投野由紀夫

英語学研究2015-Spring

本日やること

- 判別分析の概要
- NICE のデータを R に読み込む
- 判別分析の基礎的なやり方を練習する
- アウトプットの解釈の基礎を学ぶ
- その他の判別分析の手法を概観する

判別分析とは

- 人間の膨大なデータ処理：
  - 識別（discrimination）
  - 分類（classification）
  - 認識（recognition）
- 判別分析
  - コンピューターに記憶させたデータと識別すべきデータとの一致度をなんらkのモデルによって計算する、最も古典的な方法
- 線形判別分析
- 非線形判別分析
- ２群 vs. 多群の判別分析

- 回帰分析：外的基準（従属変数）は量的データ
- 判別分析では外的基準は質的データで、今回の母語話者（NS）と非母語話者（NNS）のような名義尺度であることが多い。
- 機械学習的には外的基準と用いた学習を supervised learning（教師あり学習）という。

線形判別分析（Linear Discriminant Analysis）

- グループ分けの境界が直線（説明変量が2個の場合）あるいは超平面（4個以上の場合）、次のような線形関数を用いてグループの所属を判別する：

判別関数＝ $a_0 + a_1x_1 + a_2x_2 + \ldots + a_nx_n$

データ処理の実際

- NICE のコーパス・ファイルをエッセイごとに集計：
  - Type
  - Token
  - TTR
  - Sentence number
  - Average sentence length
  - Average word length

- これらの算出は、Sketch Engine ではできないので、WordSmith5 または Casual Conc (for Mac) などで個別語彙統計を出す機能を使う

データの型

CLASS	Type	Token	TTR	STTR	SentNum	AveSenL	AveWL
NS	249	494	50.4	50.4	30	16.46666667	4.64
NS	219	465	47.1	47.1	35	13.28571429	5.26
NS	302	826	36.56	36.56	39	21.17948718	4.42
NS	273	612	44.61	44.61	26	23.53846154	4.69
NS	236	512	46.09	46.09	21	24.38095238	4.6
NS	248	504	49.21	49.21	23	21.91304348	4.85
NNS	228	488	46.72	46.72	19	25.68421053	4.16
NNS	270	610	44.26	44.26	24	25.41666667	4.98
NNS	272	576	47.22	47.22	28	20.57142857	5.2
NNS	266	566	47	47	20	28.3	5.15
NNS	219	525	41.71	41.71	24	21.875	4.3
NNS	288	620	46.45	46.45	29	21.37931034	4.83

統計ソフトの利用

- 市販：SPSS, Statistica, SAS　→ 高機能だが値段が高い

- 現在：多くの人が R を利用している

- 無料
- 常に新しいヴァージョンと統計パッケージが更新されている
- 強力なグラフィック機能がある

R のスクリプトの説明

```
library(MASS)
NICE.DATA <- read.delim(file.choose()) #データの読み込み

#NS, NNS それぞれ100サンプルずつランダムに抽出
select <- c(sample(x=1:342, size=100, replace=FALSE),
sample(x=343:552, size=100, replace=FALSE))
nice <- NICE.DATA[select,]

#単位が大小あるので、対数変換する
nice.log <- log(nice[,2:8])
nice.class <- nice[,1]
nice2 <- data.frame(nice.class, nice.log)

#linear discriminant analysis のコマンド lda()
Z.lda <- lda(nice.class ~ .,data=nice2) # Use all the variables
```

判別分析の基本コマンド

>library(MASS)

>lda(y ~ x1+x2+ … , data=DataFrame)

y: 目的変数（外的基準）

x1, x2, … : 説明変数

DataFrame : データフレームオブジェクト

判別分析のアウトプット

```
> z.lda
Call:
lda(nice.class ~ ., data = nice2)

Prior probabilities of groups:
NNS  NS
0.5 0.5

Group means:
        Type    Token      TTR     STTR  SentNum  AveSenL     AveWL
NNS 5.385142 6.410862 3.579459 3.587190 3.111274 3.299588 1.488130
NS  5.591906 6.367588 3.829481 3.831914 3.396992 2.970597 1.488481

Coefficients of linear discriminants:
                LD1
Type     -45.364921
Token     32.581500
TTR       60.970724
STTR      -5.044373
SentNum   18.103391
AveSenL   16.172400
AveWL     -1.219351
```

第１判別関数（判別軸）の係数
これを使って、判別関数を構築する

線形判別係数 → 判別関数

```
Coefficients of linear discriminants:
                LD1
Type      -45.364921
Token      32.581500
TTR        60.970724
STTR       -5.044373
SentNum    18.103391
AveSenL    16.172400
AveWL      -1.219351
```

定数項は以下のコマンドで得られる

apply(Z.lda$means%*%Z.lda$scaling,2,mean)

LD1
274.134

f_{LD1} = -45.37 x (TYPE) + 32.58 x (Token) + 60.9 x (TTR) – 5.04 x (STTR)
+ 18.10 x (SentNum) + 16.17 x (AveSenL) -1.22 x (AveWL) + 274.134

判別得点

- 判別関数で得られた1つ1つのファイルの判別得点を出力するには、predict()関数を使う

predict(z.lad)$x

これで今回ランダムに選定した200ファイルのそれぞれの判別得点が出力される

```
> apply(z.lda$means%*%z.lda$scaling,2,mean)
     LD1
274.134
> predict(z.lda)$x
              LD1
219 -0.90589290
53  -2.14957994
280 -2.28682030
250 -1.81422733
300 -1.00125924
299 -1.71136150
115 -2.27716056
50  -2.93697959
172 -0.58782084
256 -1.32775960
177 -1.36342052
33  -2.13769252
85  -1.62307879
306 -0.99944752
196 -3.07736471
167 -0.94741764
276  0.88261998
68   4.15257972
```

負が第1群＝NNS,
正が第2群＝NS

各個体の群への判別結果

>predict(Z.lda)$class

200のサンプルがこの判別関数をもとに
どちらに割り振られたかがわかる

```
> predict(Z.lda)$class
  [1] NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NS  NNS NNS NNS
 [21] NNS NNS NNS NS  NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS
 [41] NNS NNS NNS NS  NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS NNS
 [61] NNS NNS NNS NNS NS  NNS NNS NNS NNS NNS NS  NS  NNS NNS NNS NS  NNS NNS NNS NNS
 [81] NNS NNS NS  NNS NNS NNS NNS NNS NNS NNS NS  NS  NNS NNS NNS NNS NNS NNS NNS NNS
[101] NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NNS NS  NS  NNS NNS NS  NS  NS
[121] NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NNS NS  NS  NS  NS  NS  NS
[141] NS  NS  NS  NS  NS  NNS NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS
[161] NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NNS NS  NS  NS  NS
[181] NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS  NS
Levels: NNS NS
```

学習データの判別結果

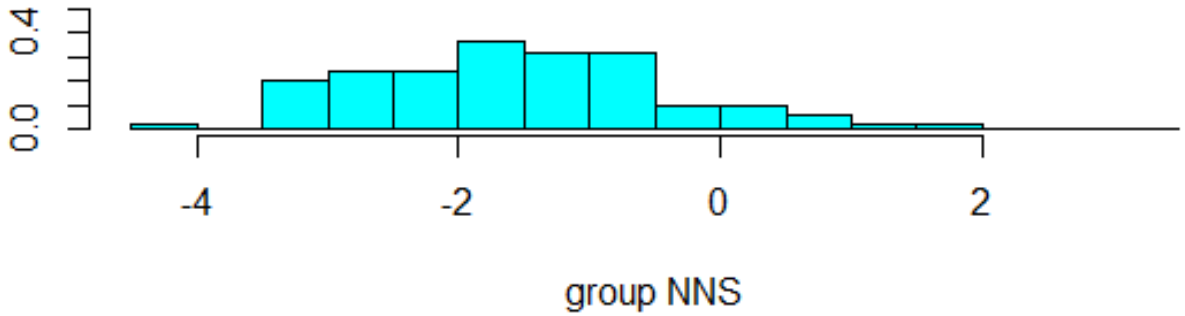
>table(nice.class,predict(Z.lda)$class)

このコマンドで学習データにおける判別結果をクロス表で確認できる

```
> table(nice.class,predict(Z.lda)$class)

nice.class NNS NS
       NNS  90 10
       NS    6 94
>
>
```

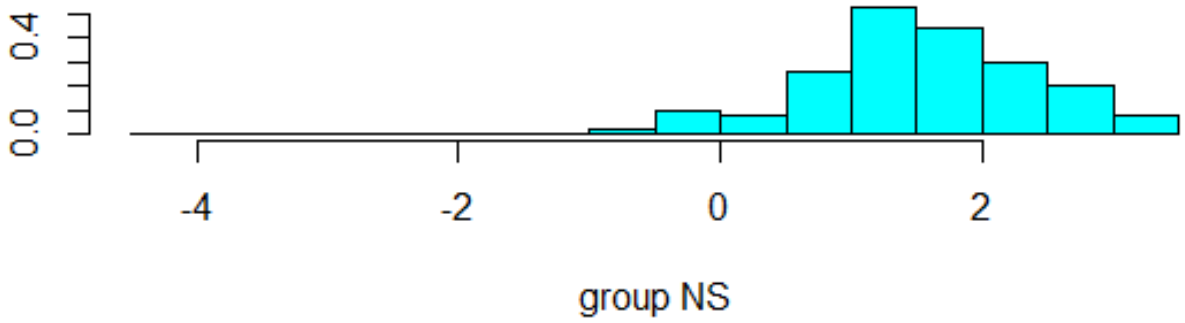
学習データの第１判別関数得点の分布

>plot(Z.lda, dimen=1)



重なる領域が多いほど
誤判別率が高くなる

判別関数が2つ以上出
た場合には散布図も
かける



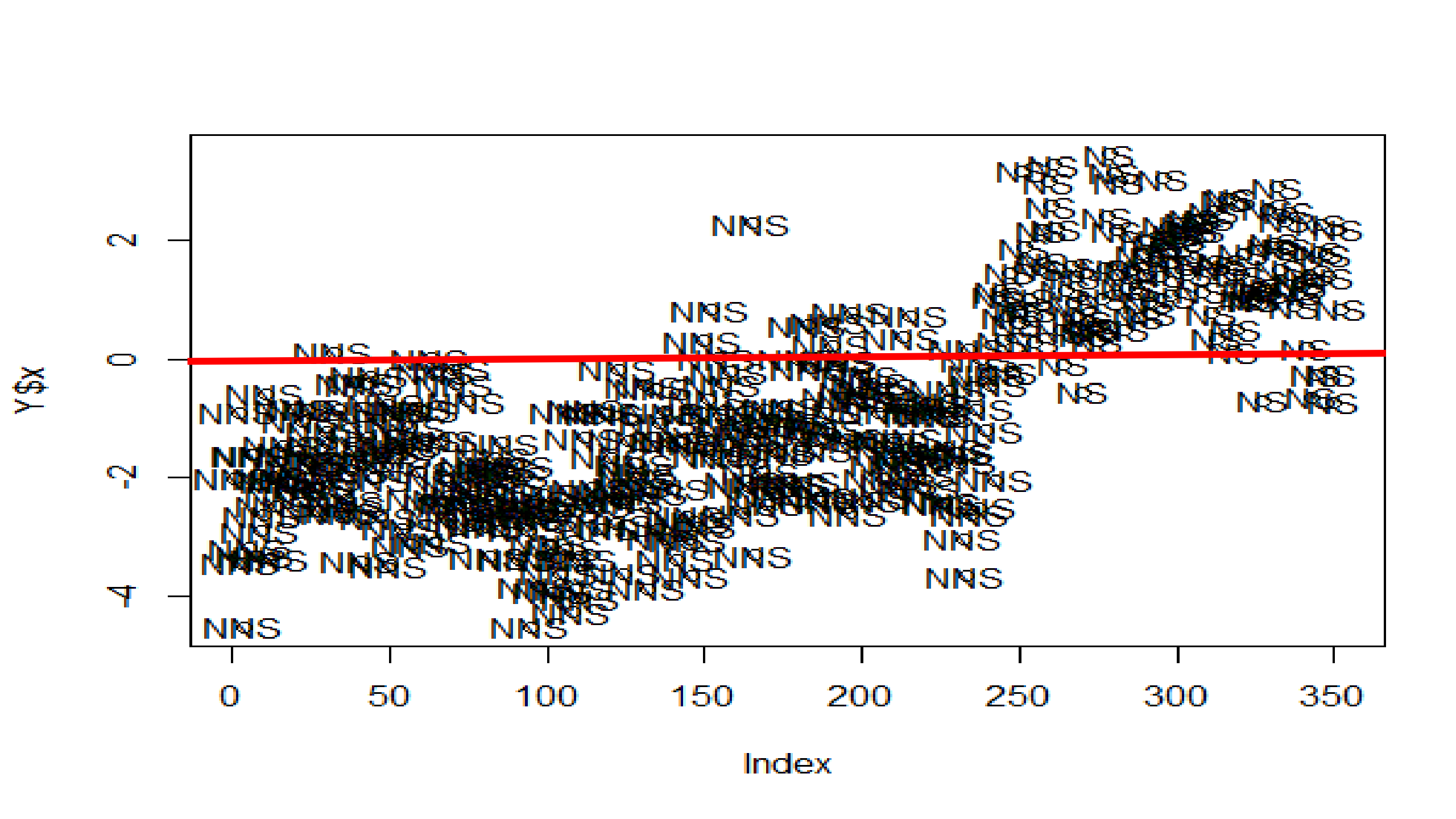
テストデータの判別

```
# prepare the test data
nice.test <- NICE.DATA[-select,]          #先ほどランダムに選択したもの以外を選んで nice.test に格納

#use logarithmic values
nice.test.log <- log(nice.test[,2:8])     #対数処理
class2 <- nice.test[,1]
nice2.test <- data.frame(class2, nice.test.log)   #新しいデータフレーム nice2.test に格納

#Use predict() functions for the test data (-1 means excluding CLASS variable)
Y <- predict(Z.lda,nice2.test[,-1] )      Z.ldaの結果を使って予測

#show the confusion matrix
table(nice2.test[,1], Y$class)
```

予測結果

- 線形判別式で残りのデータもかなりの精度で判別できた
- 誤判別率は　NNS　＝　5.4%　NS = 7.2%

```
> Y <- predict(Z.lda,nice2.test[,-1] )
> table(nice2.test[,1], Y$class)

     NNS  NS
NNS 229  13
NS    8 102
```


Y$x
Index
2
0
-2
-4
0
50
100
150
200
250
300
350
NNS

判別分析のその他の手法

- 線形判別分析
  - データが多変量正規分布に従い、グループの母分散が等しいという仮定がある
  - あまりに独立変数が多いと向いていない

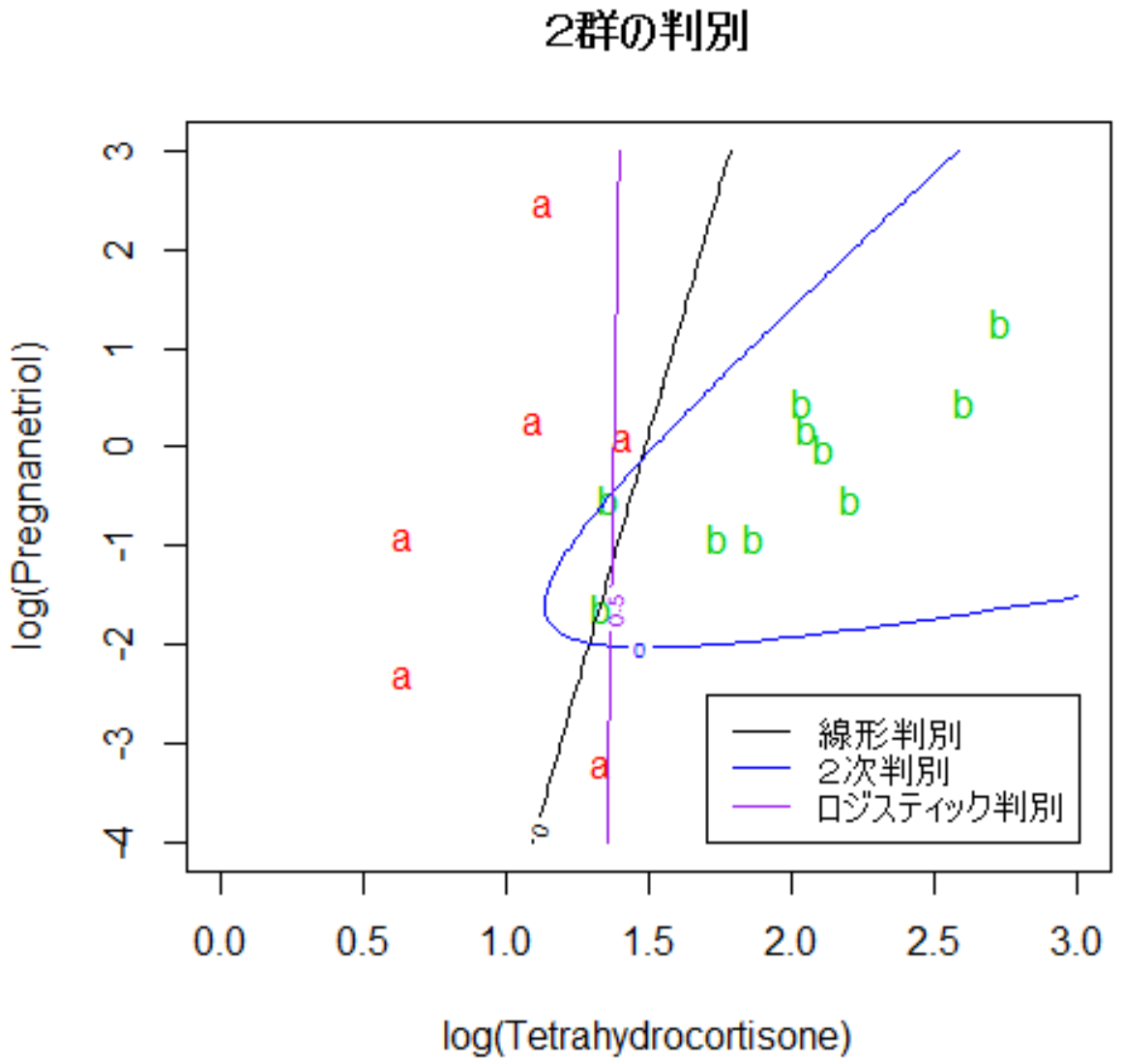
- 非線形判別分析（non-linear discriminatory analysis）
  - 2次(quadratic)式を用いる　２次判別分析 qda()
- k 最近傍法（k-Nearest Neighbor）
  - 判別すべき固体の周辺の最も近い k個（距離は普通ユークリッド距離）を見つけてその k個の多数決でどちらのグループに属するかを決める。記憶ベース推論の手法の1つ
- ベイズ判別法（NaiveBayes）

２群の判別

- 線形判別
- ２次判別
- ロジスティック判別
  - ２値データの場合→ロジスティック回帰
  - ２つの群 A，B を記述する変数が m 個あり，B が生起する確率を p，A が生起する確率を $1-p$ とすると，ロジスティック回帰モデルで生起確率が記述できる．ここで，$p > 0.5$ なら B，$p < 0.5$ なら A と判別 すると決めれば判別分析が行える．判別の境界線は $p = 0.5$ となる直線である．R では，glm() 関数でロジスティック判別が行える．
    ロジスティック判別では，各群の確率分布を記述する必要がないので，正規母集団モデルに適合しないようなデータにも適用できる．

$$\log \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \cdots + \beta_m x_m$$

オッズの対数＝ロジット関数



東京大学　大森宏　先生のページ
http://lbm.ab.a.u-tokyo.ac.jp/~omori/kensyu/discriminant.htm