

データマイニング

第7回 線形判別分析

2023年春学期

宮津和弘

本日の講義・演習

日付	講義・演習内容
04/14/23	(1) イントロダクション
04/21/23	(2) ビジネスシミュレーション
04/28/23	(3) ID-POSデータ分析
05/12/23	(4) 対応分析
05/19/23	(5) クラスター分析
05/26/23	(6) 自己組織化マップ
06/02/23	(7) 線形判別分析
06/09/23	(8) 非線形判別分析
06/16/23	(9) ツリーモデル
06/23/23	(10) 集団学習
06/30/23	(11) サポートベクターマシン
07/04/23	(12) ネットワーク分析
07/14/23	(13) 共分散構造分析
07/21/23	(14) テキスト分析
07/28/23	(15) まとめ



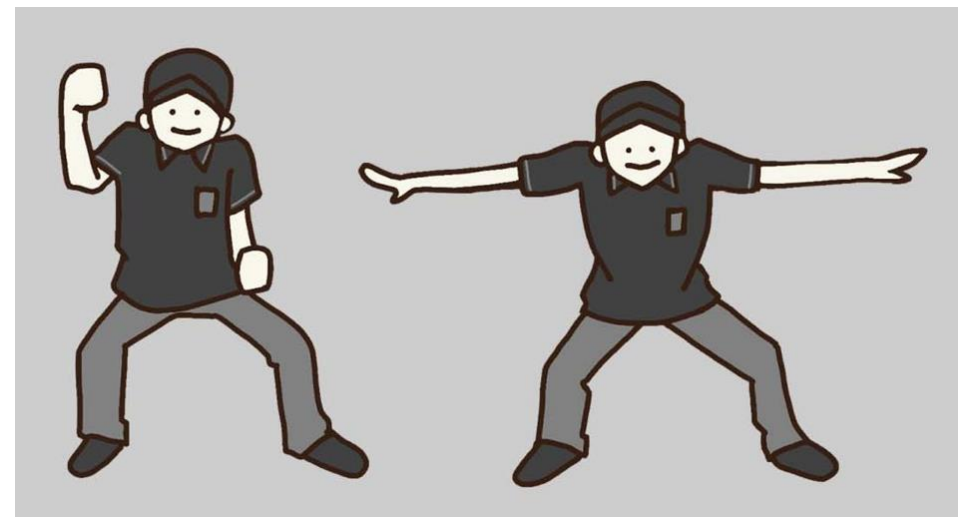
本日の演習概要とポイント

■ 機械学習と線形判別

→ 偽1000フラン札に対する2群線形判別

■ 線形判別の演習

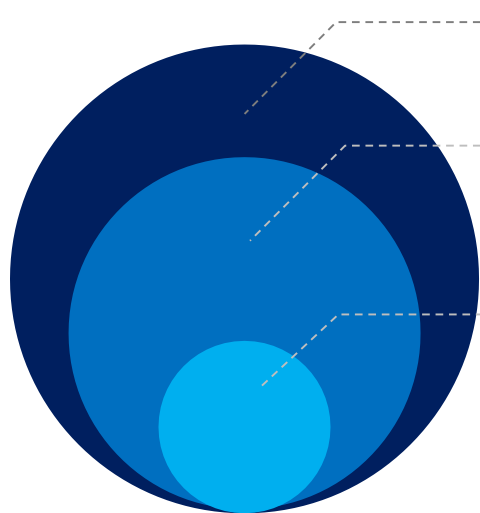
→ IRISデータを用いた多群線形判別



機械学習と線形判別

機械学習の始まりは「判別」から

機械学習では、データサイエンスにおけるデータ分析手法の一つで、データに含まれる特徴量を用いて、背景にある現象のパターンを発見・定量化して、その関係式から事象を**判別**して将来を予測を行う。



人工知能(AI) : 決まった作業や汎用的な作業を**人間に代わって**実行する

機械学習(ML) : 明示的に指示を与えなくても**自律的に**学習能力を獲得する
→ 教師あり学習、教師なし学習、強化学習

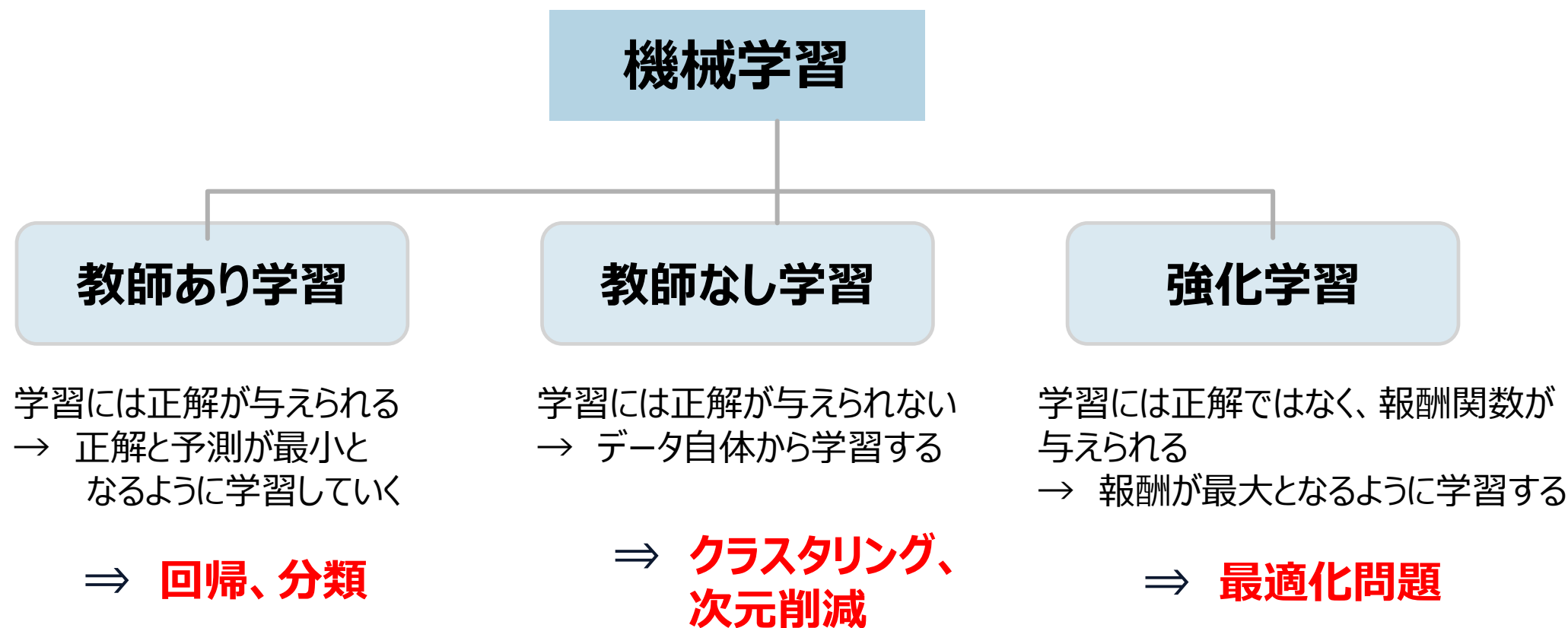
深層学習(DL) : データから自動的に特徴を**Neural Network**で学習する
→ DNN, CNN, RNN

機械学習 vs. 統計モデル

アプローチ	特長	分析データサイズ
統計モデル	<ul style="list-style-type: none">観測されない<u>背後の構造理解</u>が目的であるデータサイズは大きい推定モデルの解釈が可能である不確定要素はモデルに含まれる	サンプルデータ (～数十メガバイト)
機械学習	<ul style="list-style-type: none"><u>現象の出現を予測</u>することが目的であるデータサイズが大きい推定モデルの解釈が不可 (Black Box)<u>不確定要素は明示的に含めない</u>	全数データ (～数ペタバイト)

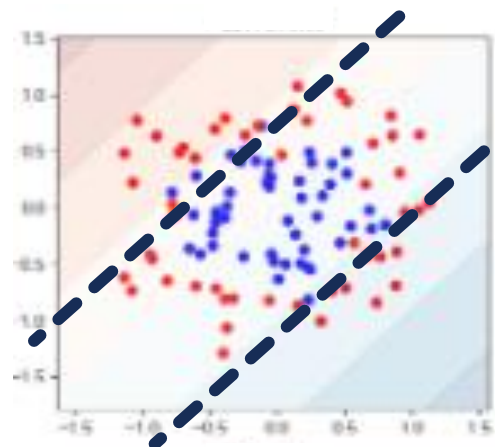
→ 機械学習はデータ依存

機械学習における学習タイプ

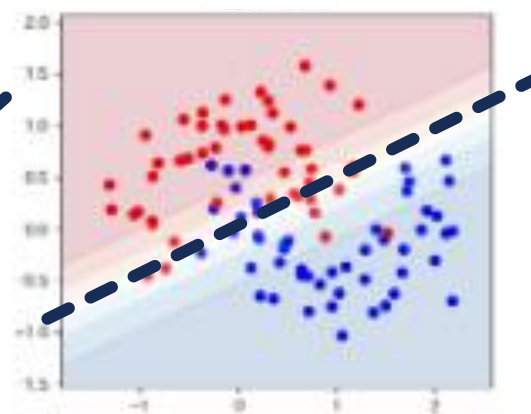


データ判別の境界

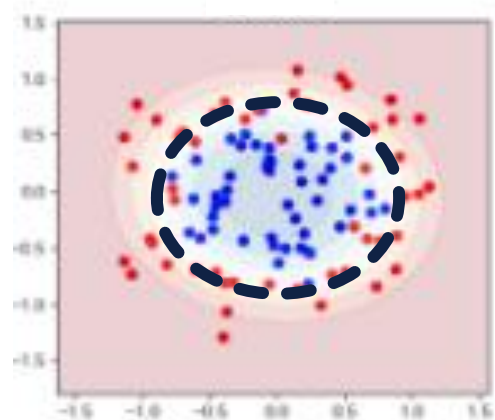
判別境界を**線形**で表現



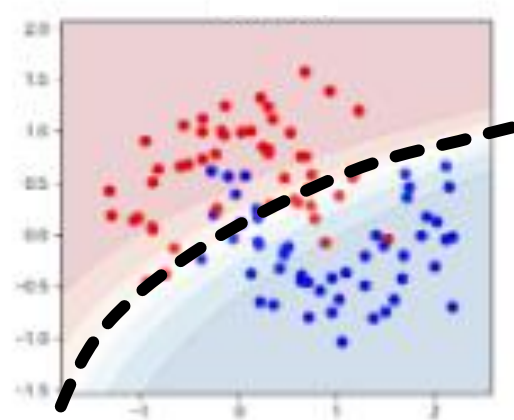
判別境界を**線形**で表現



判別境界を**二次形式**で表現



判別境界を**非線形**で表現



フィッシャーの線形判別法とは？

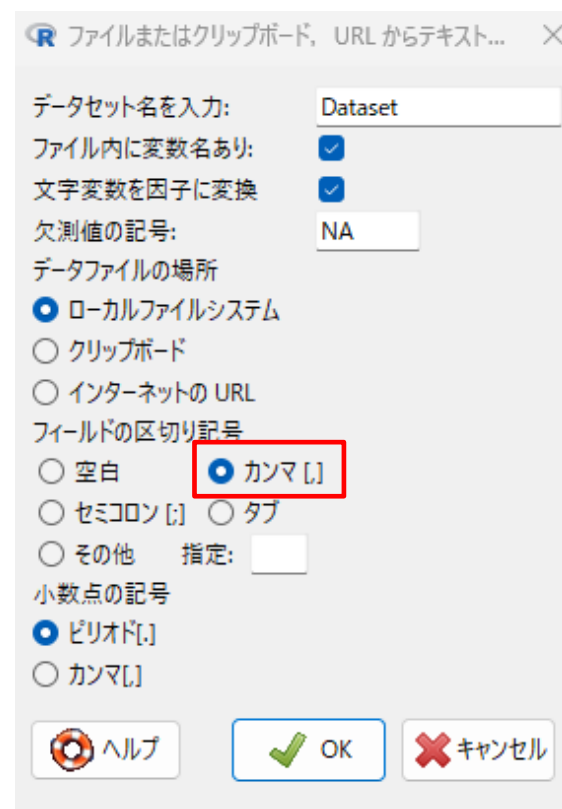
フィッシャーの線形判別法では、データが多変量正規分布に従い、各群の母分散が等しいという仮定の下、群間の分散と群内の分散の比を最大かするように係数： $\boldsymbol{\beta}$ を決定する。

$$(\text{判別関数}) \quad f_{discr} = \boldsymbol{x}^t \boldsymbol{\beta} \quad \boldsymbol{x} = (x_1, x_2, \dots, x_N)^t \quad \boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_N)^t$$

$$\left. \begin{array}{l} \text{群間分散} \quad BS = \frac{1}{g-1} \sum_{i=1}^g m_i (\bar{X}_i - \bar{X})(\bar{X}_i - \bar{X})^t \\ \text{群内分散} \quad WS = \frac{1}{m-g} \sum_{i=1}^g \sum_{j=1}^{m_i} (X_{ij} - \bar{X}_i)(X_{ij} - \bar{X}_i)^t, m = \sum_{i=1}^g m_i \end{array} \right\} V = \frac{BS}{WS} = \frac{\boldsymbol{\beta}^t (BS) \boldsymbol{\beta}}{\boldsymbol{\beta}^t (WS) \boldsymbol{\beta}}$$

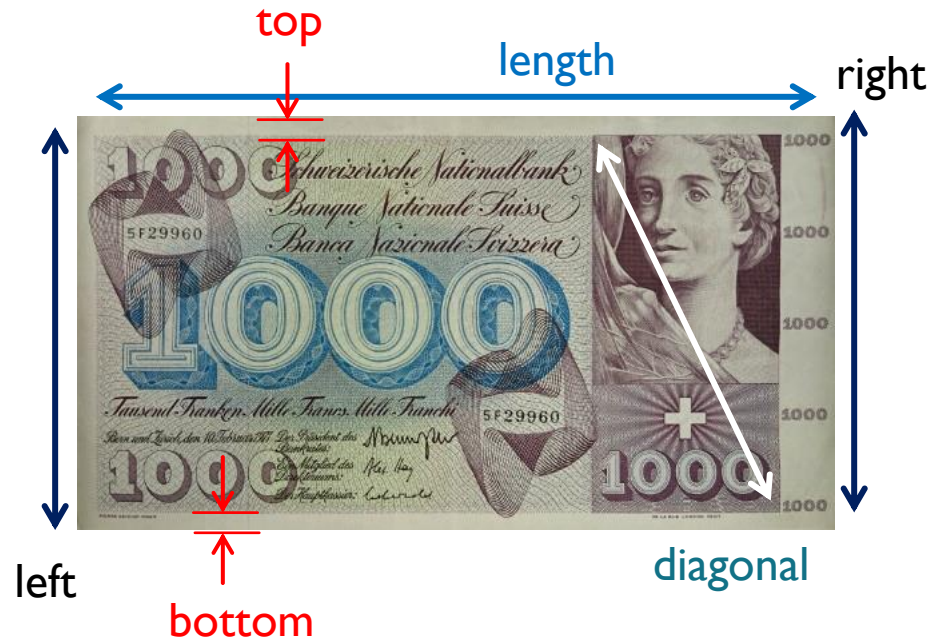
演習データの読み込み

- ① Rstudio起動する
- ② `> library(Rcmdr)` ※コマンドラインから Rコマンダー を起動する
- ③ 演習ファイル “sbnote.csv” を読み込む
 - Rstudio `> Dataset<-read.csv(“sbnote.csv”)`
又は
 - Rコマンダー (データ) → (データインポート) → (テキストファイルまたはクリップボード...) →
✓ OKを選択して、sbnote.csv を指定する
- ④ 演習データが Dataset に読み込まれる



スイス銀行偽紙幣のデータ

1000フラン紙幣の尺度指標



length	left	right	bottom	top	diagonal	class
214.8	131	131.1	9	9.7	141	0
214.6	129.7	129.7	8.1	9.5	141.7	0
214.8	129.7	129.7	8.7	9.6	142.2	0
214.8	129.7	129.6	7.5	10.4	142	0
215	129.6	129.7	10.4	7.7	141.8	0
215.7	130.8	130.5	9	10.1	141.4	0
215.5	129.5	129.7	7.9	9.6	141.6	0
214.5	129.6	129.2	7.2	10.7	141.7	0
214.9	129.4	129.7	8.2	11	141.9	0
215.2	130.4	130.3	9.2	10	140.7	0
215.3	130.4	130.3	7.9	11.7	141.8	0
215.1	129.5	129.6	7.7	10.5	142.2	0
215.2	130.8	129.6	7.9	10.8	141.4	0
214.7	129.7	129.7	7.7	10.9	141.7	0
215.1	129.9	129.7	7.7	10.8	141.8	0
214.5	129.8	129.8	9.3	8.5	141.6	0
214.6	129.9	130.1	8.2	9.8	141.7	0
215	129.9	129.7	9	9	141.9	0
215.2	129.6	129.6	7.4	11.5	141.5	0
214.7	130.2	129.9	8.6	10	141.9	0

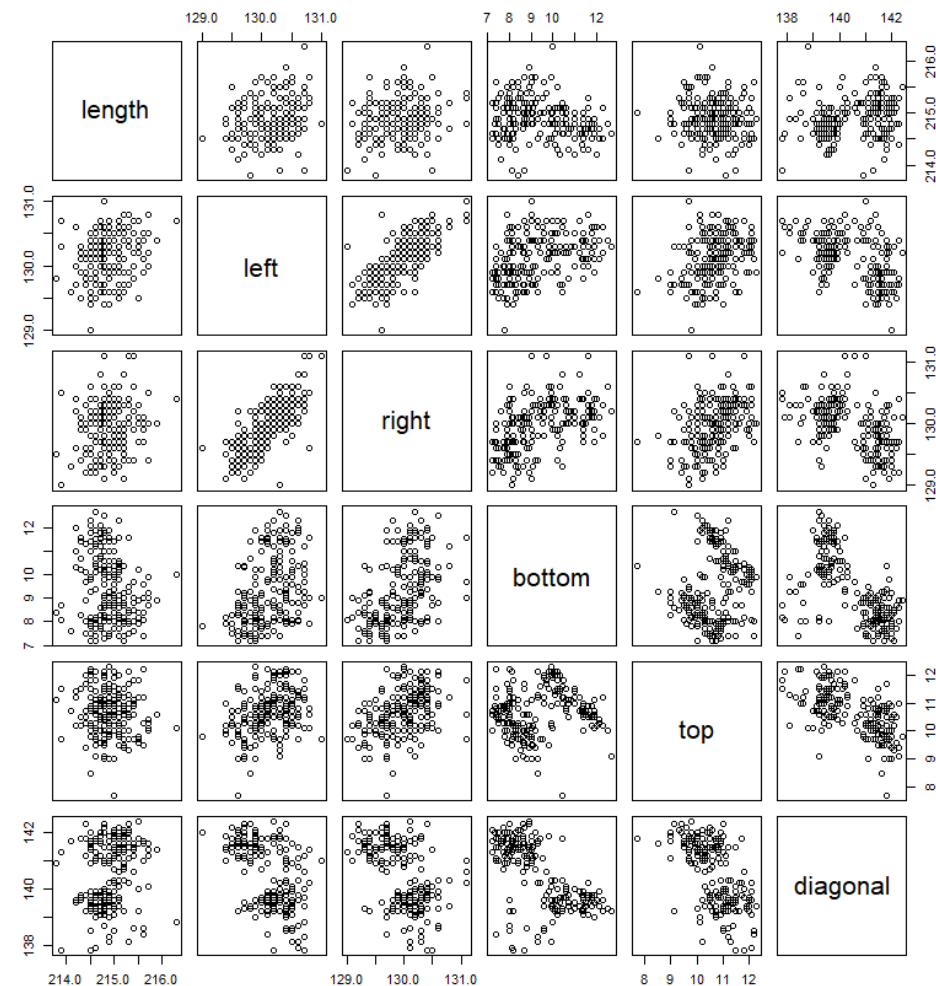
真偽判定：1（偽札）、0（真札）

1000フラン紙幣データの散布図

```
> plot(Dataset[,1:6])  
> cor(Dataset[,1:6])
```

	length	left	right	bottom	top	diagonal
length	1.0000000	0.2427039	0.1679278	-0.1838481	-0.06644058	0.1878632
left	0.24270392	1.0000000	0.7432628	0.4133781	0.36234960	-0.5032290
right	0.16792779	0.7432628	1.0000000	0.4864230	0.40067021	-0.5164755
bottom	-0.18384810	0.4133781	0.4864230	1.0000000	0.14194006	-0.6229711
top	-0.06644058	0.3623496	0.4006702	0.1419401	1.0000000	-0.5940446
diagonal	0.18786320	-0.5032290	-0.5164755	-0.6229711	-0.5940446	1.0000000

- 左右の長さの相関が最も高い
- 対角線長さはその他の長さとの負の相関

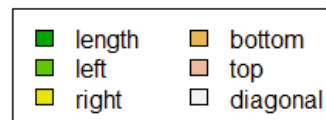
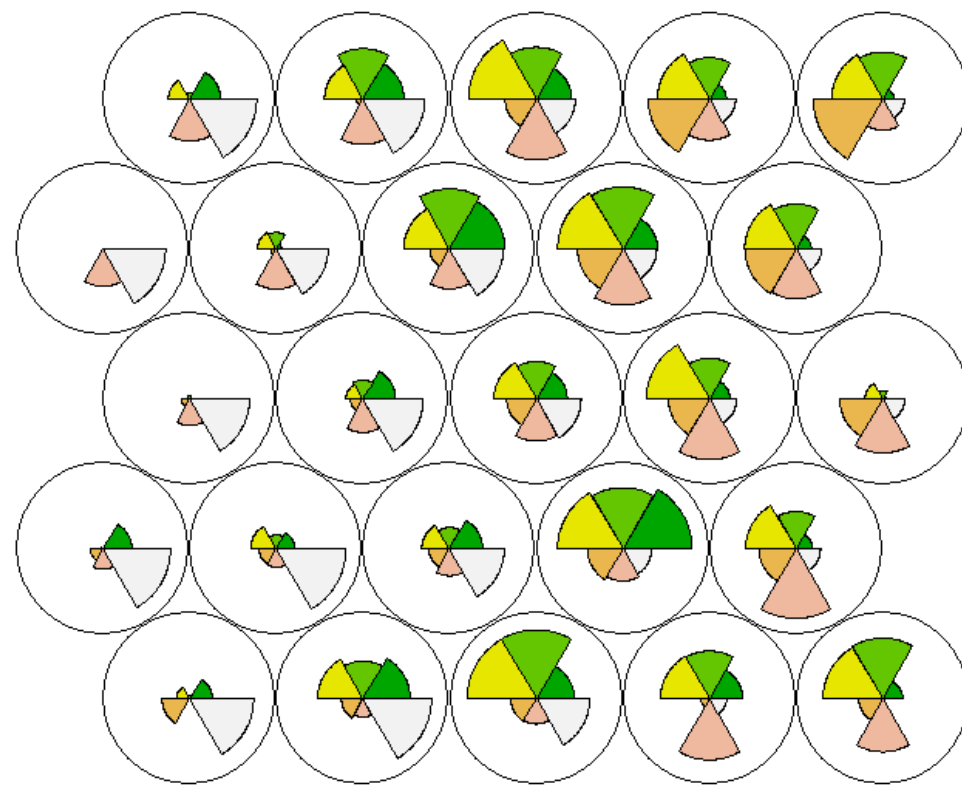


自己組織化マップで視覚化してみよう！

```
> gr <- somgrid(topo="hexagonal", xdim=5,ydim=5)  
> note.som <- som(as.matrix(Dataset[,1:6]),gr,rlen=200)  
> plot(note.som)
```

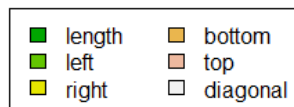
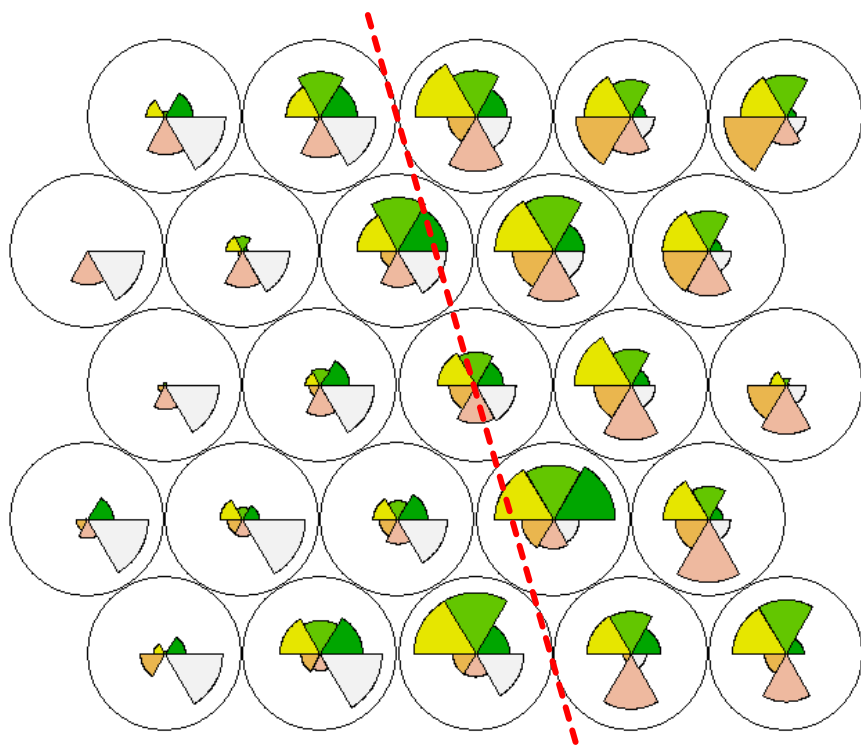
5 × 5 グリッドに 5 つの特徴量で
自己組織化マップを出力

→ 丸の中の扇の大きさは各特徴量の
大きさを表す！

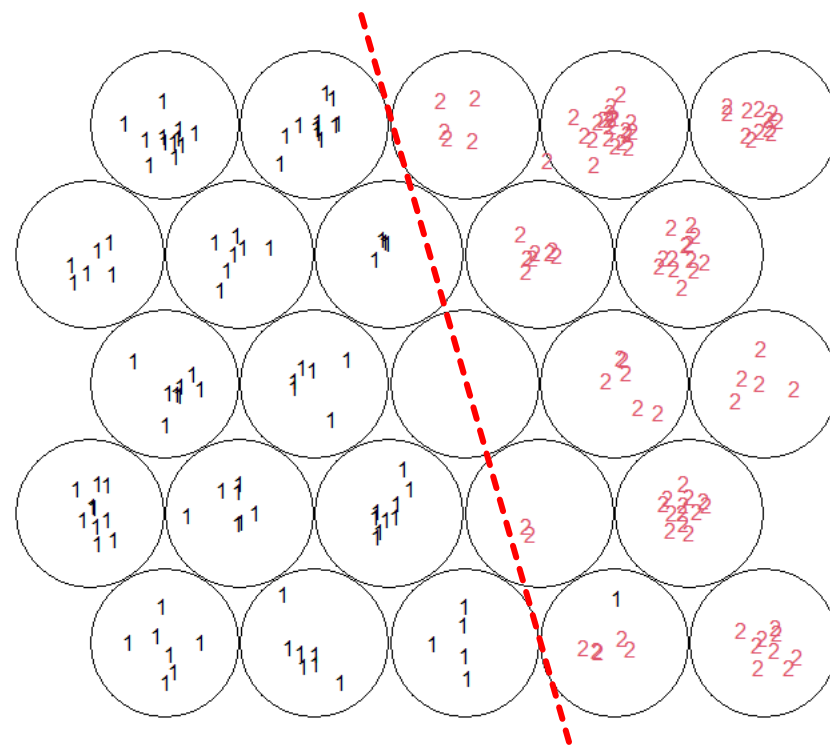


自己組織化マップとラベルの対応づけ

```
> lab.cod1 <- as.numeric(as.factor(Dataset[,7]))  
> plot(note.som, type="mapping", labels=lab.cod1, col=lab.cod1)
```



偽札の方がサイズが少し大きい!?



1: 真札, 2: 偽札

線形判別モデルで偽札を見つけ出そう！

MASSパッケージの **lda** (linear discriminant analysis) コマンドを利用する

```
> library(MASS)
> even.n <- 2*(1:100)-1
> Dataset.training <- Dataset[even.n,]
> Dataset.test <- Dataset[-even.n,]
```

偽札データの偶数番目データを **学習データ**
偽札データの奇数番目データを **検証データ**

```
> Z.lda <- lda(class~.,data=Dataset.training)
> table(Dataset.training[,7], predict(Z.lda)$class)
```

判別モデルの結果は **Z.lda** に格納される！

	0	1
0	50	0
1	0	50

学習データで構築した判別モデルでは、
学習データの全てを正しく判別できている！

学習データの判別結果

判別モデルの構築

```
> z.lda
call:
lda(class ~ ., data = Dataset.train)
```

Prior probabilities of groups:

```
  0  1
0.5 0.5
```

Group means:

	length	left	right	bottom	top	diagonal
0	215.014	129.934	129.744	8.320	10.236	141.498
1	214.826	130.284	130.184	10.504	11.158	139.552

Coefficients of linear discriminants:

	LD1
length	0.4798261
left	-0.8285631
right	0.7297746
bottom	1.1457880
top	1.0787731
diagonal	-1.7453162

特徴量係数

```
> apply(z.lda$means%*%z.lda$scaling,2,mean)
```

LD1
-132.7716

定数項

$$f = 0.480x_1 - 0.829x_2 + 0.730x_3 + 1.146x_4 + 1.079x_5 - 1.745x_6 + 132.77$$

判別モデルによる検証データの評価

検証データに対する判別を実行

```
> y<-predict(Z.lda, Dataset.test[, -7])  
> table(Dataset.test[, 7], y$class)
```

	0	1
0	49	1
1	0	50

検証データの判別結果



1000フラン紙幣の真偽確率

```
> y$posterior
```

	(真札) 0	(偽札) 1
2	1.000000e+00	3.952555e-15
4	1.000000e+00	8.958861e-16
6	9.999999e-01	5.334394e-08
8	1.000000e+00	2.505646e-15
10	9.996440e-01	3.560466e-04
12	1.000000e+00	6.849452e-15
14	1.000000e+00	7.271813e-12
16	1.000000e+00	6.641068e-14
18	1.000000e+00	1.254354e-14
20	1.000000e+00	1.615036e-13
22	1.000000e+00	6.067543e-12
24	1.000000e+00	5.535109e-10
26	1.000000e+00	5.535611e-13
28	1.000000e+00	7.557365e-10
30	1.000000e+00	3.470711e-18
32	1.000000e+00	1.560234e-17
34	9.999998e-01	2.365683e-07
36	1.000000e+00	9.855710e-11
38	1.000000e+00	6.589325e-16
40	1.000000e+00	2.092477e-16

90	1.000000e+00	9.603729e-14
92	1.000000e+00	9.692612e-11
94	1.000000e+00	8.274691e-13
96	1.000000e+00	1.054738e-10
98	1.000000e+00	4.902528e-11
100	1.000000e+00	3.330830e-13
102	1.575609e-13	1.000000e+00
104	5.609591e-05	9.999439e-01
106	2.047481e-08	1.000000e+00
108	1.191366e-08	1.000000e+00
110	7.274141e-08	9.999999e-01
112	1.460508e-09	1.000000e+00
114	8.204957e-11	1.000000e+00
116	2.877359e-05	9.999712e-01
118	2.024303e-11	1.000000e+00
120	5.336640e-13	1.000000e+00
190	3.033607e-13	1.000000e+00
192	1.095079e-10	1.000000e+00
194	1.047724e-12	1.000000e+00
196	8.516900e-12	1.000000e+00
198	4.426242e-09	1.000000e+00
200	2.916702e-10	1.000000e+00

判別モデル器

	length	left	right	bottom	top	diagonal
214.8	214.8	131	131.1	9	9.7	141
214.8	214.8	129.7	129.7	8.1	9.5	141.7
214.8	214.8	129.7	129.7	8.7	9.6	142.2
214.8	214.8	129.7	129.6	7.5	10.4	142
215	215	129.6	129.7	10.4	7.7	141.8
215.7	215.7	130.8	130.5	9	10.1	141.4
215.5	215.5	129.5	129.7	7.9	9.6	141.6
214.5	214.5	129.6	129.2	7.2	10.7	141.7
214.9	214.9	129.4	129.7	8.2	11	141.9
215.2	215.2	130.4	130.3	9.2	10	140.7
215.3	215.3	130.4	130.3	7.9	11.7	141.8
215.1	215.1	129.5	129.6	7.7	10.5	142.2
215.2	215.2	130.8	129.6	7.9	10.8	141.4

length →

left →

right →

bottom →

top →

diagonal →

線形判別器

→ 真札の確率

→ 偽札の確率

	0	1
2	1.000000e+00	3.952555e-15
4	1.000000e+00	8.958861e-16
6	9.999999e-01	5.334394e-08
8	1.000000e+00	2.505646e-15
10	9.996440e-01	3.560466e-04
12	1.000000e+00	6.849452e-15
14	1.000000e+00	7.271813e-12
16	1.000000e+00	6.641068e-14
18	1.000000e+00	1.254354e-14
20	1.000000e+00	1.615036e-13
22	1.000000e+00	6.067543e-12
24	1.000000e+00	5.535109e-10
26	1.000000e+00	5.535611e-13
28	1.000000e+00	7.557365e-10
30	1.000000e+00	3.470711e-18
32	1.000000e+00	1.560234e-17
34	9.999998e-01	2.365683e-07
36	1.000000e+00	9.855710e-11
38	1.000000e+00	6.589325e-16
40	1.000000e+00	2.092477e-16

判別モデルによる検証データの可視化

```
> plot(Y$x,type="n")  
> text(Y$x,labels=Dataset.test$class)
```

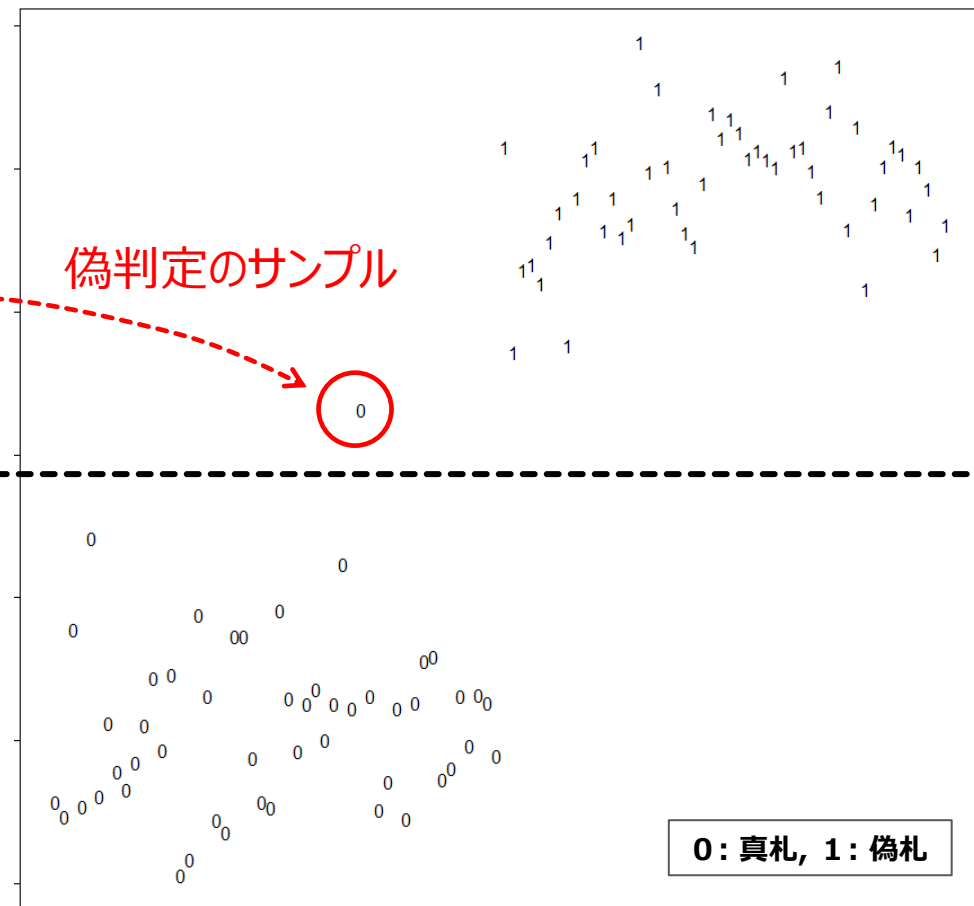


検証データの判別結果

	0	1
0	49	1
1	0	50

偽判定のサンプル

0



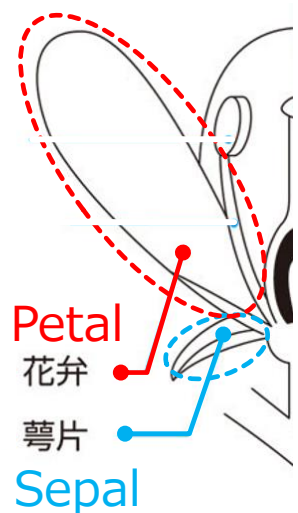


多群線形判別



IRISデータについて

アヤメの種類によって、
花弁と**がく片**の幅と長さ
で分類するためのデータ
として用いられる



Sepal.Length	Sepal.width	Petal.Length	Petal.width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor
5.5	2.3	4.0	1.3	versicolor
6.5	2.8	4.6	1.5	versicolor
5.7	2.8	4.5	1.3	versicolor
6.3	3.3	4.7	1.6	versicolor
4.9	2.4	3.3	1.0	versicolor
6.6	2.9	4.6	1.3	versicolor
5.2	2.7	3.9	1.4	versicolor
6.3	3.3	6.0	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7.1	3.0	5.9	2.1	virginica
6.3	2.9	5.6	1.8	virginica
6.5	3.0	5.8	2.2	virginica
7.6	3.0	6.6	2.1	virginica
4.9	2.5	4.5	1.7	virginica
7.3	2.9	6.3	1.8	virginica
6.7	2.5	5.8	1.8	virginica
7.2	3.6	6.1	2.5	virginica



IRISデータの読み込み

Rコマンダーから 【データ】 → 【パッケージ内のデータ】 → 【アタッチされたパッケージからデータセットを…】



データセット名に“**iris**”入力してOKすると、
irisデータが読み込まれるのを確認（↓）

		1 2 3 4 5				
	rowname	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	1	5.1	3.5	1.4	0.2	setosa
2	2	4.9	3.0	1.4	0.2	setosa
3	3	4.7	3.2	1.3	0.2	setosa
4	4	4.6	3.1	1.5	0.2	setosa
5	5	5.0	3.6	1.4	0.2	setosa
6	6	5.4	3.9	1.7	0.4	setosa
7	7	4.6	3.4	1.4	0.3	setosa
8	8	5.0	3.4	1.5	0.2	setosa
9	9	4.4	2.9	1.4	0.2	setosa
10	10	4.9	3.1	1.5	0.1	setosa
11	11	5.4	3.7	1.5	0.2	setosa
12	12	4.8	3.4	1.6	0.2	setosa

IRISを3分類の線形判別をしてみよう！

rowname	1	2	3	4	5
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3.0	1.4	0.1	setosa
14	4.3	3.0	1.1	0.1	setosa
15	5.8	4.0	1.2	0.2	setosa
16	5.7	4.4	1.5	0.4	setosa
17	5.4	3.9	1.3	0.4	setosa
18	5.1	3.5	1.4	0.3	setosa
19	5.7	3.8	1.7	0.3	setosa
20	5.1	3.8	1.5	0.3	setosa

50	5.0	3.3	1.4	0.2	setosa
51	7.0	3.2	4.7	1.4	versicolor
52	6.4	3.2	4.5	1.5	versicolor
53	6.9	3.1	4.9	1.5	versicolor
54	5.5	2.3	4.0	1.3	versicolor
55	6.5	2.8	4.6	1.5	versicolor
56	5.7	2.8	4.5	1.3	versicolor
57	6.3	3.3	4.7	1.6	versicolor

100	5.7	2.8	4.1	1.3	versicolor
101	6.3	3.3	6.0	2.5	virginica
102	5.8	2.7	5.1	1.9	virginica
103	7.1	3.0	5.9	2.1	virginica
104	6.3	2.9	5.6	1.8	virginica
105	6.5	3.0	5.8	2.2	virginica
106	7.6	3.0	6.6	2.1	virginica
107	4.9	2.5	4.5	1.7	virginica
108	7.3	2.9	6.3	1.8	virginica
109	6.7	2.5	5.8	1.8	virginica
110	7.2	3.6	6.1	2.5	virginica
111	6.5	3.2	5.1	2.0	virginica
112	6.4	2.7	5.3	1.8	virginica
113	6.8	3.0	5.5	2.1	virginica
114	5.7	2.5	5.0	2.0	virginica
115	5.8	2.8	5.1	2.4	virginica



実際には、**学習用データ**と**テストデータ**に分けて実行し、構築した線形判別モデルの評価まで行う

学習データと検証データに分割

```
> data("iris")
> iris.lab<- c(rep("s",50),rep("c",50),rep("v",50))
> iris1<-data.frame(iris[,1:4],iris.lab)
> iris.train <- iris1[even.n,]
> iris.test  <- iris1[-even.n,]
```

学習データサンプルに
偏りのないように抽出

※ 学習データサンプルによって構築される判別モデルに
差異が生じるため、学習サンプルを複数準備して
判別モデル係数の平均値を採用することもある！



Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.8	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor
5.5	2.3	4.0	1.3	versicolor
6.5	2.8	4.6	1.5	versicolor
5.7	2.8	4.5	1.3	versicolor
6.3	3.3	4.7	1.6	versicolor
4.9	2.4	3.3	1.0	versicolor
6.6	2.9	4.6	1.3	versicolor
5.2	2.7	3.9	1.4	versicolor
5.0	2.0	3.5	1.0	versicolor
5.9	3.0	4.2	1.5	versicolor
6.0	2.2	4.0	1.0	versicolor
6.1	2.9	4.7	1.4	versicolor
5.6	2.9	3.6	1.3	versicolor
6.3	3.3	6.0	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7.1	3.0	5.9	2.1	virginica
6.3	2.9	5.6	1.8	virginica
6.5	3.0	5.8	2.2	virginica
7.6	3.0	6.6	2.1	virginica
4.9	2.5	4.5	1.7	virginica
7.3	2.9	6.3	1.8	virginica
6.7	2.5	5.8	1.8	virginica
7.2	3.6	6.1	2.5	virginica
6.5	3.2	5.1	2.0	virginica
6.4	2.7	5.3	1.9	virginica
6.8	3.0	5.5	2.1	virginica
5.7	2.5	5.0	2.0	virginica
5.8	2.8	5.1	2.4	virginica

線形判別関数の推定

```
> library(MASS)
> z.lda<-lda(iris.lab~.,data=iris.train)
> z.lda
Call:
lda(iris.lab ~ ., data = iris.train)
```

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

```
> apply(z.lda$means%*%z.lda$scaling,2,mean)
```

	LD1	LD2
	1.486146	6.282412

$$f = -0.592x_1 - 1.842x_2 + 1.653x_3 + 3.563x_4 - 1.486$$

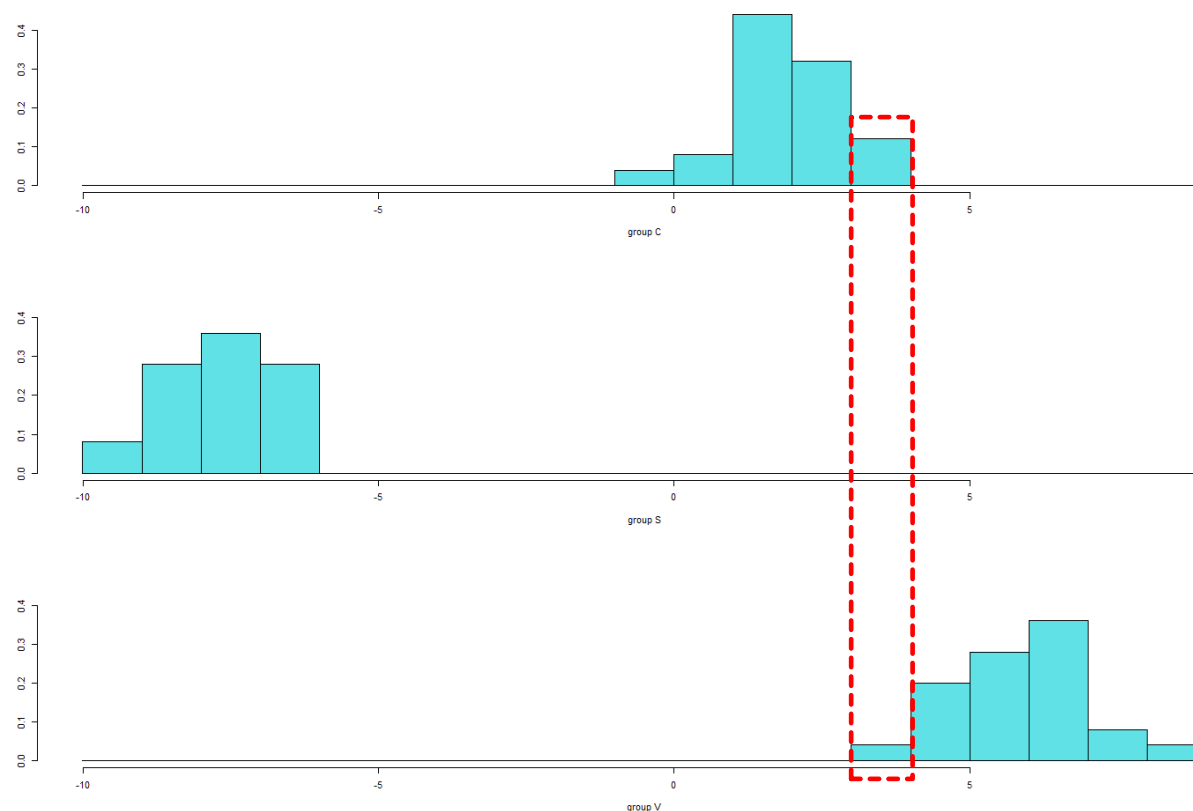
学習データの判別評価

```
> table(iris.train[,5],predict(z.lda)$class)
```

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

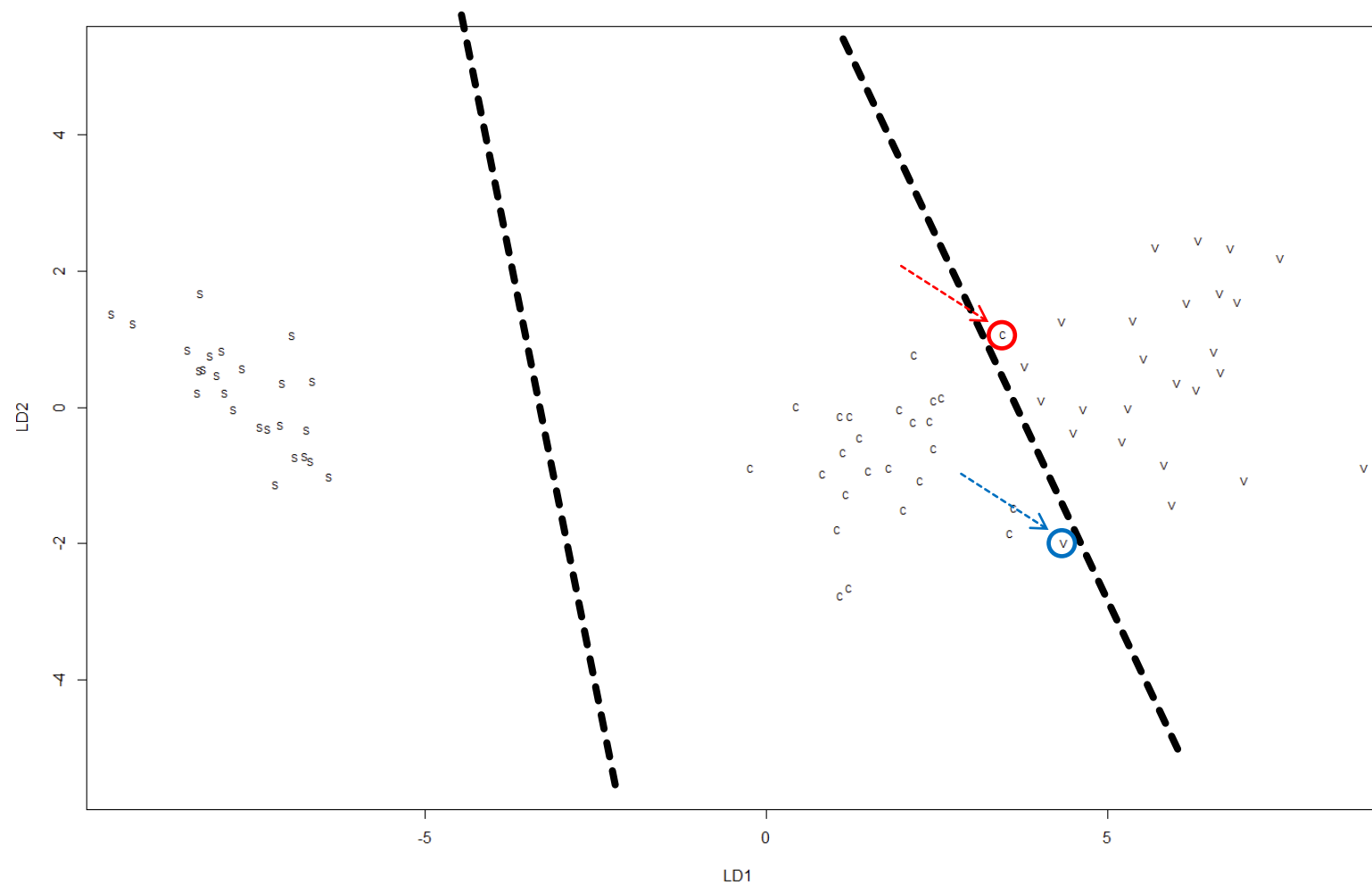
```
>  
> plot(z.lda,dimen=1)
```

※ **C** と **V** で一部オーバーラップしている！



学習データの判別関数得点

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

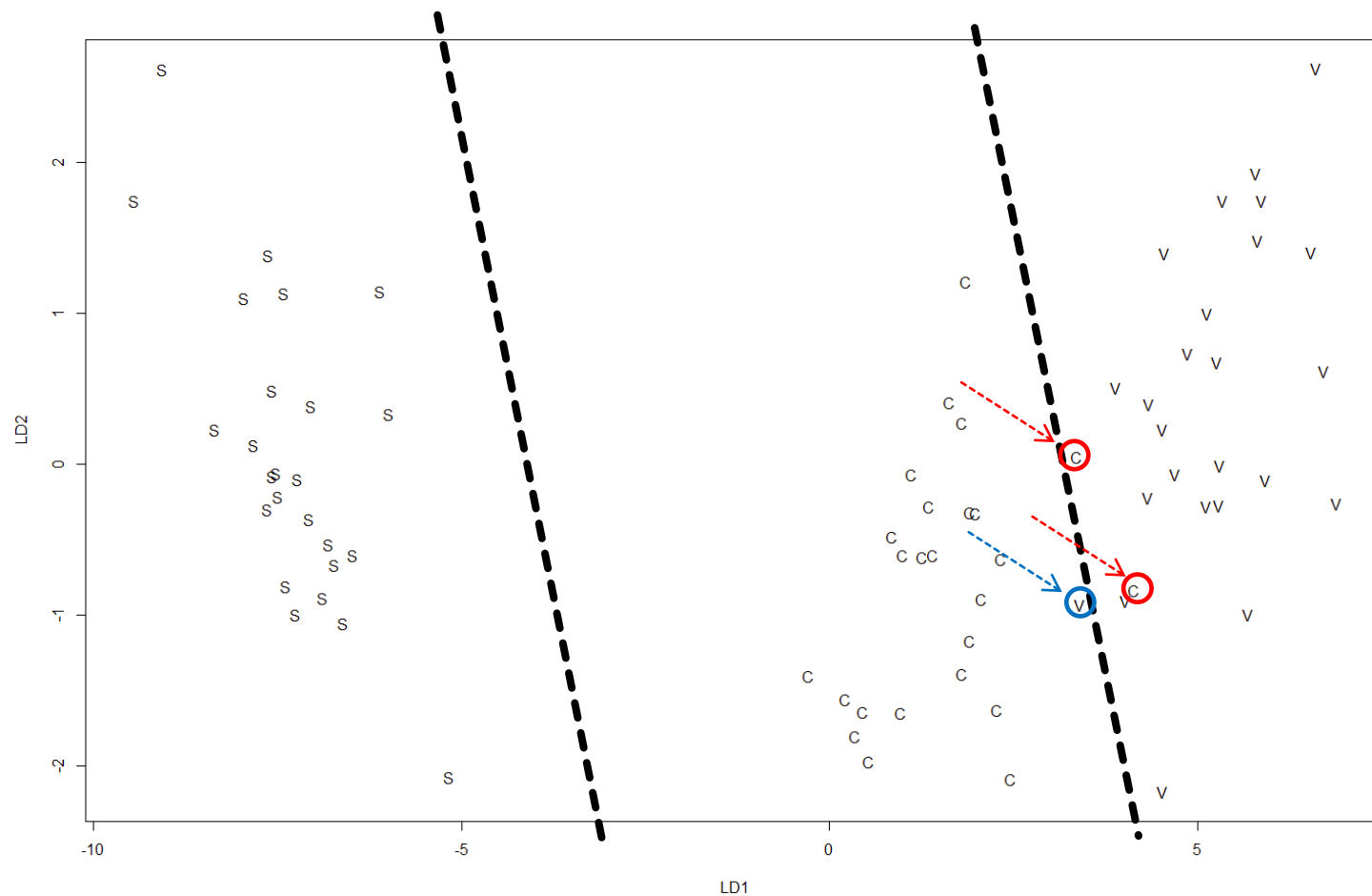


検証データの判別結果

```
> table(iris.test[,5],y$class)
```

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

```
> plot(y$x,type="n")  
> text(y$x,labels=iris.test$iris.lab)
```



課題：SPAMデータを用いた線形判別

SPAM1で提供された7つの特徴量を用いて、ldaコマンドを用いて線形判別モデルを構築・評価しなさい。

SPAMとHAMの各サンプルから半分ずつを学習データとして線形判別モデルを構築し、残り半分を検証データを用いて判別能力を評価・考察しなさい。



迷惑メール

vs.



通常メール

	remove	free	email	you	charExclamation	capitalAve	capitalTotal	type
1	0.00	0.32	1.29	1.93	0.778	3.756	278	spam
2	0.21	0.14	0.28	3.47	0.372	5.114	1028	spam
3	0.19	0.06	1.03	1.36	0.276	9.821	2259	spam
4	0.31	0.31	0.00	3.18	0.137	3.537	191	spam
5	0.31	0.31	0.00	3.18	0.135	3.537	191	spam
6	0.00	0.00	0.00	0.00	0.000	3.000	54	spam
7	0.00	0.96	0.32	3.85	0.164	1.671	112	spam
8	0.00	0.00	0.00	0.00	0.000	2.450	49	spam
9	0.30	0.00	0.15	1.23	0.181	9.744	1257	spam
10	0.38	0.00	0.12	1.67	0.244	1.729	749	spam
11	0.96	0.00	0.96	3.84	0.462	1.312	21	spam
12	0.25	0.00	0.00	1.16	0.663	1.243	184	spam
13	0.00	0.34	1.39	2.09	0.786	3.728	261	spam
14	0.90	0.00	0.00	2.72	0.000	2.083	25	spam
15	0.00	5.35	0.00	3.21	0.357	1.971	205	spam
16	0.42	1.27	0.00	1.70	0.572	5.659	249	spam
17	0.00	0.00	0.00	1.88	0.428	4.652	107	spam
18	0.00	0.00	0.00	0.00	1.975	35.461	461	spam
19	0.18	0.00	0.37	3.15	0.455	1.320	70	spam
20	0.00	0.63	3.18	2.22	0.055	3.509	186	spam



データマイニングを楽しもう！