Rによるデータマイニング

山本義郎
東海大学理学部
yamamoto@sm.u-tokai.ac.jp

本日の内容

- ？？？？？
- Rとは
- Rによる解析の実践
  - コマンドと関数の利用
  - データの入力、インポート
  - データの要約
- Rによるデータマイニング
  - 多変量解析
  - データマイニングとは
  - データマイニング手法

サマーセミナーとスプリングセミナー

- サマーセミナー
  - 大山(1993)
  - 広島(1995?)
  - 屋久島(1998)
  - 支笏湖(1999)
  - 嬬恋(2003)
- スプリングセミナー
  - 指宿(1997)
  - 有馬温泉(1998)
  - 河口湖 (1999)
  - 因島

続きは明日の夜のセッションで

関数電卓としてRを使う

- 算術演算子
  - +, -, *, /, ^ (べき乗)
- 数学関数および統計関数

```
> 2+2
[1] 4
> 2^3
[1] 8
> (1-2)*3
[1] -3
> 1-2*3
[1] -5
>
```

```
> sqrt(2)    # the square root
[1] 1.414214
> sin(pi)    # the sine function, pi is constant
[1] 1.224606e-16
> exp(1)     # exp(x) = e^x
[1] 2.718282
> log(10)    # the log base e
[1] 2.302585
>
```


関数電卓としての利用 (2)

- 付値
  - “=” または “<-”

```
> x=2
> x+3
[1] 5
> e^2
Error: object "e" not found
> e=exp(1)
> e^2
[1] 7.389056
>
```

  - 変数名の制限

    英数で始まり、演算子 (+, -, *, /)を含まない名前。日本語もOK

変数 (データベクトル)

- c( ) 関数
- 例
  - 1990年代にテキサスの海岸に打ち上げられたクジラの数

    74, 122, 235, 111, 292, 111, 211, 133, 156, 79

    付値

```
> whales.texas = c(74,122,235,111,292,111,211,133,156,79)
>
```


変数への関数の適用

```
> sum(whales.texas)   # total number of beaching
[1] 1524
> length(whales.texas) # length of data vector
[1] 10
> sum(whales.texas)/length(whales.texas) # average
[1] 152.4
> mean(whales.texas)   # mean function finds average
[1] 152.4

> sort(whales.texas)   # sorted data
 [1]  74  79 111 111 122 133 156 211 235 292
> min(whales.texas)   # the minimum value
[1] 74
> max(whales.texas)   # the maximum value
[1] 292
> range(whales.texas) # range return both min and max
[1]  74 292
> cumsum(whales.texas) # running tally
 [1]   74  196  431  542  834  945 1156 1289 1445 1524
```


ライブラリの利用

- Rには必要最低限の関数、データが用意されている
- ＋αはパッケージとして提供されている
  - パッケージの利用 library(パッケージ名)
  - データの利用 data(データ名)
  - インストール install.packages(パッケージ名)
- Windows版では

「ヘルプ」の参照

- コマンド
  - help( ) 関数
  - 関数の検索
    - help.search(), apropos()
  - help.start()
  - example( ) 関数
  - history( ) 関数
- Windows, Mac「ヘルプ」メニューから

データの入力

- Rのデータの型と観測尺度
  - 質的変数
    - 名義尺度 (factor)
    - 順序尺度 (ordered)
  - 量的変数
    - 間隔尺度・比率尺度 （numeric・・integer, single, double）
- データの編集
  - edit()
  - data.entry()

Cars93データ

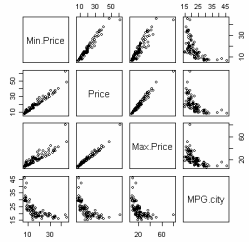
```
> library(MASS)
> data(Cars93)                    data(Cars93,package="MASS")
> summary(Cars93)
    Manufacturer     Model        Type      Min.Price        Price
 Chevrolet: 8    100    : 1   Compact:16   Min.   : 6.70   Min.   : 7.40   最小値
 Ford     : 8    190E   : 1   Large  :11   1st Qu.:10.80   1st Qu.:12.20   第1四分位数
 Dodge    : 6    240    : 1   Midsize:22   Median :14.70   Median :17.70   平均値
 Mazda    : 5    300E   : 1   Small  :21   Mean   :17.13   Mean   :19.51   中央値
 Pontiac  : 5    323    : 1   Sporty :14   3rd Qu.:20.30   3rd Qu.:23.30   第3四分位数
 Buick    : 4    535i   : 1   Van    : 9   Max.   :45.40   Max.   :61.90   最大値
 (Other)  :57    (Other):87
```

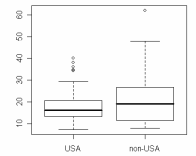
質的変数　　量的変数

```
> attach(Cars93)
> table(Origin,DriveTrain)
        DriveTrain
Origin    4WD Front Rear
  USA       5    34    9
  non-USA   5    33    7
```


グラフの作成

```
> plot(Cars93[,4:7])
> hist(Cars93$Price)
> boxplot(Price~Origin)
```





回帰直線

$y = \beta_0 + \beta_1 x$　　y の x による(y のx上への) 回帰方程式

$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad (i = 1,2,\cdots,n)$

誤差項の性質　$E(\varepsilon_i) = 0, V(\varepsilon_i) = \sigma^2, Cov(\varepsilon_i, \varepsilon_j) = 0$

最小2乗法による回帰係数の推定

残差平方和　$S(\beta_0, \beta_1) = \sum \{y_i - (\beta_0 + \beta_1 x_i)\}^2$　を最小にする β_0, β_1

$$\begin{cases} \dfrac{\partial S}{\partial \beta_0} = \dfrac{\partial}{\partial \beta_0} \sum \{y_i - (\beta_0 + \beta_1 x_i)\}^2 = 0 \\ \dfrac{\partial S}{\partial \beta_1} = \dfrac{\partial}{\partial \beta_1} \sum \{y_i - (\beta_0 + \beta_1 x_i)\}^2 = 0 \end{cases}$$

正規方程式

$$\begin{cases} n\beta_0 + \left(\sum x_i\right)\beta_1 = \sum y_i \\ \left(\sum x_i\right)\beta_0 + \left(\sum x_i^2\right)\beta_1 = \sum x_i y_i \end{cases}$$

$$\hat{\beta}_1 = \frac{s_{xy}}{s_x^2} = \frac{r s_y}{s_x}, \quad \hat{\beta}_0 = \bar{y} - \beta_1 \bar{x}$$

Rでの単回帰分析

- 単回帰分析の出力

```
> cars.lm=lm(MPG.city~Horsepower,data=Cars93)
> summary(cars.lm)

Call:
lm(formula = MPG.city ~ Horsepower, data = Cars93)

Residuals:
   Min     1Q Median     3Q    Max
-7.879 -2.529 -1.033  2.358 17.223

Coefficients:                                    回帰係数の検定統計量のp値
             Estimate Std. Error t value Pr(>|t|)   （この結果から、回帰係数は有意）
(Intercept) 32.746279   1.273229  25.719  < 2e-16 ***
Horsepower  -0.072174   0.008323  -8.671 1.54e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.181 on 91 degrees of freedom
Multiple R-Squared: 0.4524,     Adjusted R-squared: 0.4464
F-statistic: 75.19 on 1 and 91 DF,  p-value: 1.537e-13
```

回帰係数

重回帰分析

$y = \beta_0 + \beta_1 x_i + \beta_2 x_2 + \cdots + \beta_p x_p$　　y の $x_1, x_2, .., x_p$ による回帰方程式

$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_p x_{pi} + \varepsilon_i \quad (i = 1,2,\cdots,n)$

誤差項の性質　$E(\varepsilon_i) = 0, V(\varepsilon_i) = \sigma^2, Cov(\varepsilon_i, \varepsilon_j) = 0$

最小自乗法による回帰係数の推定

残差平方和　$S(\beta_0, \beta_1, \cdots, \beta_p) = \sum \{y_i - (\beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi})\}^2$

モデルの妥当性の検定

- 分散分析表

要因	平方和	自由度	平均平方	F値
回帰	SS_R	p	$V_R = SS_R/p$	$F = V_R/V_E$
残差	SS_E	$n-(p+1)$	$V_E = SS_E/(n-p-1)$	
全体	SS_T	$n-1$		

F 統計量は，モデルによる変動と残差の変動が同じであるという仮説の下で、自由度p, N-p-1のF分布に従う

- 回帰係数の検定

仮説　$H_0 : \beta_i = 0 \quad vs \quad H_1 : \beta_i \neq 0 \qquad t_i = \dfrac{\hat{\beta}_i - \beta_i}{s.e.(\hat{\beta}_i)} \sim t_{n-2} \quad (i = 1,\cdots,p)$

Rでの重回帰分析

```
> plot(Cars93[c(7:8,12:15)])
> cor(Cars93[c(7:8,12:15)])
> cars.lm2=lm(MPG.city~Horsepower+EngineSize+RPM,data=Cars93)
> summary(cars.lm2)

Call:
lm(formula = MPG.city ~ Horsepower + EngineSize + RPM, data = Cars93)

Residuals:
    Min      1Q  Median      3Q     Max
-6.2279 -1.8775  0.1103  1.4753 15.5367

Coefficients:                                    回帰係数の検定統計量のp値
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 12.337394   6.715351   1.837  0.06952 .
Horsepower  -0.077097   0.016433  -4.692 9.74e-06 ***
EngineSize   0.227510   0.991088   0.230  0.81896
RPM          0.003884   0.001174   3.307  0.00136 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.599 on 89 degrees of freedom
Multiple R-Squared: 0.6032,     Adjusted R-squared: 0.5898
F-statistic:  45.1 on 3 and 89 DF,  p-value: < 2.2e-16
```

回帰係数

重相関係数の2乗

分散分析のp値

コマンド利用とプログラムの作成

- コマンドの利用
  - 関数電卓として
  - 付値と演算
  - 統計解析関数
  - 行列とベクトルの演算

行列演算・・重回帰分析の例

$$y_i = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon_i,\ \varepsilon_i \sim N(0, \sigma^2)$$

行列で考えると

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon},\ \boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \sigma^2 I)$$

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix},\ X = \begin{pmatrix} 1 & x_{11} - \bar{x}_1 & \cdots & x_{1p} - \bar{x}_1 \\ 1 & x_{21} - \bar{x}_1 & \cdots & x_{2p} - \bar{x}_2 \\ \cdots & \cdots & & \cdots \\ 1 & x_{n1} - \bar{x}_1 & \cdots & x_{np} - \bar{x}_p \end{pmatrix},\ \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix},\ \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \vdots \\ \varepsilon_p \end{pmatrix}$$

回帰係数　$\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T \boldsymbol{y}$

行列演算・・重回帰分析の例

```
> y=Cars93$MPG.city
> x=Cars93[,12:14]
> X=cbind(1, x-mean(x))
> X=as.matrix(X)
> solve(t(X)%*%X)%*%t(X)%*%y
                    [,1]
1           13.459880374
EngineSize   0.071629173
Horsepower  -0.075248390
RPM          0.003741825
```

cbind()　・・・列の結合
t()　・・・転置
solve()　・・・逆行列

確率分布と乱数

- Families of distribution
  - d*dist*　.. *dist* 分布の密度(確率)関数 $f(x)$
  - p*dist*　.. *dist* 分布の分布関数 $F(x)$
  - q*dist*　.. *dist* 分布の分位点
  - r*dist*　.. *dist* 分布の乱数

*dist*の例
2項・・・binom
正規・・・norm
対数正規・・・lnorm
t分布・・・t
カイ2乗・・chisq
F分布・・・f

例）一様分布

```
> dunif(x=1, min=0, max=3)
> dunif(1,0,3)
> punif(q=2,0,3)
> qunif(p=1/2,0,3)
> runif(n=1,0,3)
> runif(5,0,3)
```

例）正規分布

```
> dnorm(1.96)
> pnorm(1.96)
> qnorm(0.975)
> rnorm(5)
> rnorm(5,mean=60,sd=8)
```


シミュレーション

- 標本抽出

```
> (urand=runif(10,0,3))
> sample(urand,size=5, replace=TRUE)
> sample(urand,size=5, replace=FALSE)
> sample(urand,size=10, replace=FALSE)
```

```
> sample(c(0,1), size=20, replace=T) # 20 times coin toss
> (x=sample(c(0,1), size=20, replace=T, prob=c(1/4,3/4)) )
> barplot(table(x))
> par(mfcol=c(3,5))
> for (i in 1:15){
 x=sample(c(0,1), size=20, replace=T, prob=c(1/4,3/4) )
 barplot(table(x))
}
```


関数の作成

```
par(mfcol=c(1,1))
toss<-function(n){
x=sample(c(0,1), size=n, replace=T)
barplot(table(x))
}
toss(10)

toss10<-function(n){
x=sample(c(0,1), size=10, replace=T)
meanx=mean(x)
for (i in 2:n){
  x=c(x,sample(c(0,1), size=100, replace=T))
  meanx=c(meanx,mean(x))
}
plot(meanx,type="l",main="コイン投げ(10枚x回数)の表の出現率の推移")
}
toss10(100)
```


多変量解析手法

主成分分析

```
> plot(iris)
> iris.pca = princomp(iris[,1:4], cor=T) # PCA for correlation matrix
> iris.pca
> summary(iris.pca)
> plot(iris.pca)
> iris.pca$loadings
> plot(iris.pca$scores[,1:2],type="n",main="PC scores",xlab="pc1", ylab="pc2")
> text(iris.pca$scores[,1:2],iris.pca)
```

クラスター分析

```
> iris.hc = hclust(dist(iris[,1:4]), method="ward")
> plclust(iris.hc)
> cutree(iris.hc,3)
> table(cutree(iris.hc,3),iris$Species)
```


主成分　線形結合

$$\begin{aligned} z_1 &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{1p}x_p \\ z_2 &= a_{21}x_1 + a_{22}x_2 + \cdots + a_{2p}x_p \\ &\vdots \qquad\qquad \vdots \\ z_q &= a_{q1}x_1 + a_{q2}x_p + \cdots + a_{qp}x_p \end{aligned}$$

q 変数で説明

$$\vdots \qquad\qquad \vdots$$

$$z_p = a_{p1}x_1 + a_{p2}x_2 + \cdots + a_{pp}x_p$$

$z_1, z_2, \cdots, z_p$ は互いに無相関

$$Var(z_1) > Var(z_2) > \cdots > Var(z_p)$$

主成分分析の実施の選択項目

- 共分散行列か相関行列か
  - そのままの尺度と標準化した尺度
- 主成分数
  - 寄与率：　80%が目安
  - 固有値：　1以上
  - スクリープロット：　なだらなになる前まで
  - 解釈可能：　意味のある軸

主成分分析の手順

- 分析する変数の指定
- データのタイプ
  - 共分散行列.. そのままの単位で
  - 相関行列..単位の違いをなくすため標準化
- 主成分数の決定
  - 固有値・寄与率など
- 主成分得点を使った分析
  - プロット
  - 主成分回帰

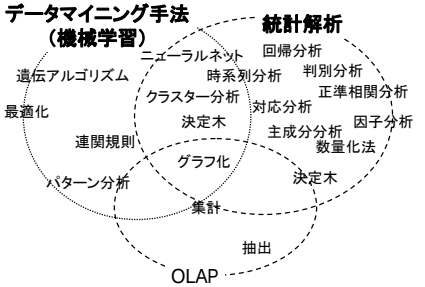
Rによる主成分分析

```
> iris.pca = princomp(iris[,1:4], cor=T)
> iris.pca
> summary(iris.pca)
Importance of components:
                          Comp.1    Comp.2     Comp.3      Comp.4
Standard deviation     1.7083611 0.9560494 0.38308860 0.143926497
Proportion of Variance 0.7296245 0.2285076 0.03668922 0.005178709
Cumulative Proportion  0.7296245 0.9581321 0.99482129 1.000000000
> plot(iris.pca)
> eqscplot(iris.pca$scores[,1:2],type="n")
> text(iris.pca$scores[,1:2],iris$species)
> iris.sp = c(rep("s",50),rep("c",50),rep("v",50))
> eqscplot(iris.pca$scores[,1:2],type="n")
> text(iris.pca$scores[,1:2],iris.sp)
```


データマイニングとは

- 大規模なデータの中から有益な関係性を見つけること
  - カンや経験から知識・ルールへ
- データマイニングと統計学
  - 多くの共通点
  - 統計学：仮説の検証
    - 妥当性の検証のためには重要
  - データマイニング：仮説の探索

データマイニング・統計手法・OLAP



データマイニング（目的と手法）

目的	手法	適用例(マーケティング)
予測 （判別予測 と数値予測）	・決定木 ・ニューラルネット ・ロジスティック回帰 ・重回帰分析（線形・非線形） ・判別分析	・スコアリング ・チャーン ・不正発見 ・リスク管理
分類 セグメンテーション	・自己組織化マップ ・クラスター分析 ・ニューラルネット ・主成分分析 ・コレスポンデンス分析 ・決定木	・優良顧客の属性 ・市場セグメンテーション
関連性の発見 （リンク分析）	・連関規則 ・時系列パターン分析 ・主成分分析 ・コレスポンデンス分析	・マーケットバスケット分析

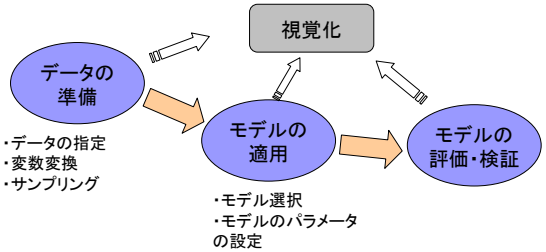
統計学の立場からのデータマイニング

- 統計学
  - 記述統計学
  - 推測統計学
    - 検定力はデータの数が大きくなると1に近づく
- 多変量データ解析
  - 分布論からの精度の評価
  - 母集団のとらえかた
  - データのスパース性
  - 非線形性
  - 最適性と一致性
- 大量データと計算機
  - シミュレーション（Bootstrap）
  - 交差妥当化（Cross Validation）

データマイニングを実施するために

- 道具（ソフトウェア）
  - 大量データを扱うことができる
  - データマイニングの基本的な手法を用意している
    - クラスター分析、決定木、ニューラルネット、他多変量解析手法
  - 視覚化（グラフ表現）機能を備えている
  - 外れ値の摘出機能を備えている
  - データベースや他のデータソースからデータをインポートできる。

データマイニングの流れ



判別（カテゴリー予測）の為の分析

- 目的
  - 複数の変数によって興味のあるカテゴリ変数の値を予測する
  - ある変数の、各カテゴリの判別における重要度を知る
- データの特徴
  - 判別分析：基準変数（カテゴリ）←説明変数（量的/ダミー）
  - ロジスティック回帰：基準変数（2値）←説明変数（量的/ダミー）
  - ニューラル、決定木：基準変数（カテゴリ）←説明変数（量的、質的）
- 利点
  - 予測と同時に各変数の相対的な影響力が分かる

ロジスティック回帰分析

$$p(Y=1)=\frac{\exp(\beta_0+\beta_1x_1+\cdots+\beta_px_p)}{1+\exp(\beta_0+\beta_1x_1+\cdots+\beta_px_p)}$$

$$\log\frac{p(Y=1)}{1-p(Y=1)}=\beta_0+\beta_1x_1+\cdots+\beta_px_p$$

回帰係数の推定には、最尤推定法が用いられる

ロジスティック回帰分析

- モデルの適合度の比較
  - 尤度比検定
- 回帰係数の検定やチェック
  - Wald 検定 (*t* 検定と同じようなもの)
    - 漸近的な性質を利用
  - オッズ比により解釈が可能(exp(β))

Rによるロジスティック回帰分析(1)

```
> data(kyphosis,package="rpart")
> summary(kyphosis)
 Kyphosis       Age             Number           Start
 absent :64   Min.   :  1.00   Min.   : 2.000   Min.   : 1.00
 present:17   1st Qu.: 26.00   1st Qu.: 3.000   1st Qu.: 9.00
              Median : 87.00   Median : 4.000   Median :13.00
              Mean   : 83.65   Mean   : 4.049   Mean   :11.49
              3rd Qu.:130.00   3rd Qu.: 5.000   3rd Qu.:16.00
              Max.   :206.00   Max.   :10.000   Max.   :18.00
> par(mfcol=c(1,3))
> boxplot(Age~Kyphosis,data=kyphosis, main="Age")
> boxplot(Number~Kyphosis,data=kyphosis, main="Number")
> boxplot(Start~Kyphosis,data=kyphosis, main="Start")
```


Rによるロジスティック回帰分析(2)

```
> kyphosis.lreg = glm(Kyphosis~.,family=binomial,data=kyphosis)
> summary(kyphosis.lreg)
Call:
glm(formula = Kyphosis ~ ., family = binomial, data = kyphosis)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3124  -0.5484  -0.3632  -0.1659   2.1613
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.036934   1.449575  -1.405  0.15996
Age          0.010930   0.006446   1.696  0.08996 .
Number       0.410601   0.224861   1.826  0.06785 .
Start       -0.206510   0.067699  -3.050  0.00229 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 83.234  on 80  degrees of freedom
Residual deviance: 61.380  on 77  degrees of freedom
AIC: 69.38
Number of Fisher Scoring iterations: 5
```


Rによるロジスティック回帰分析(3)

```
> anova(kyphosis.lreg, test="Chisq")
Analysis of Deviance Table
Model: binomial, link: logit
Response: Kyphosis
Terms added sequentially (first to last)
       Df Deviance Resid. Df Resid. Dev P(>|Chi|)
NULL                      80     83.234
Age     1    1.302        79     81.932     0.254
Number  1   10.306        78     71.627     0.001
Start   1   10.247        77     61.380     0.001
> exp(kyphosis.lreg$coef)
(Intercept)         Age      Number       Start
  0.1304281   1.0109904   1.5077239   0.8134181
> exp(kyphosis.lreg$coef[2]*12)
     Age
1.140157
> table(kyphosis$Kyphosis, round(kyphosis.lreg$fitted))
            0  1
  absent   61  3
  present 10  7
```

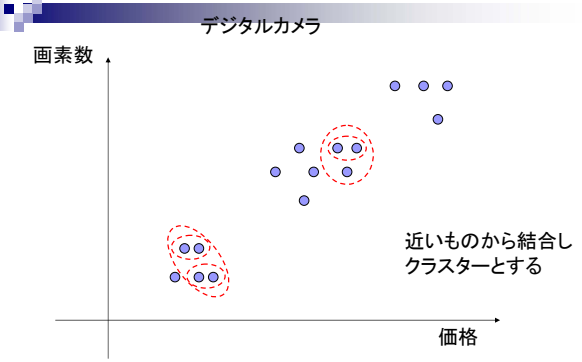

クラスター分析

- 目的
  - 複数の変数の情報から類似しているケースをグループ化する
  - 複数の変数を類似したグループにクラスタリングする
- データの特徴
  - 基準変数なし、量的・度数・2値データ
- 利点
  - 有効な分類軸が分かっていないデータを、興味のある情報に基づいてグループ化できる
  - 他の手法で得られた次元得点などによりケースをクラスタリングすることもできる
- ツール・メニュー
  - 非階層クラスタ分析:k-means法。高速。あらかじめクラスタ数を設定
  - 階層クラスタ分析:クラスタ化の方法により多種の方法がある
- 重要な出力
  - 所属クラスタ、デンドログラム(階層型の場合)
- 注意点
  - 選択する距離測定手法やクラスタ化の方法によってかなり異なる結果となりうる
  - クラスタ数の決定は恣意的

階層的クラスターの距離

- **重心法**
  - クラスターの重心の距離を用いる。
- **ウォード法**
  - 各ステップで形成される任意の2つのクラスターの平方和を最小にするようにクラスター生成を行う。
- **群平均法**
  - 2つのクラスター間の距離は、2つのクラスター内のすべてのものののペアの距離の平均して定義する
- 最近隣法
  - 2つのクラスター間の距離はそれぞれのクラスター内の最も近いもの間の距離として定義
- 最遠隣法
  - クラスター間の距離は、それぞれのクラスター内の任意の2つのもの間の距離の最大値として定義される。
- 重み付き群平均法
  - 各クラスターの大きさを重みとする点を除けば、群平均法と同じで、クラスターサイズが非常に異なる場合に適切

クラスターの作成



Rによる階層型クラスター分析

- hclust(距離データ, method="手法")
  - 距離データ dist() 関数
    - euclidean
    - maximum
    - manhattan
    - canberra
    - binary
    - minkovski
  - clusterパッケージ
    - daisy()関数
  - 手法
    - single ‥ 最近隣法
    - complete ‥ 最遠隣法
    - average ‥ 群平均法
    - centroid ‥ 重心法
    - median ‥ メディアン法
    - ward ‥ ウォード法
    - mcquitty ‥ McQuitty法

Rによる階層型クラスター分析

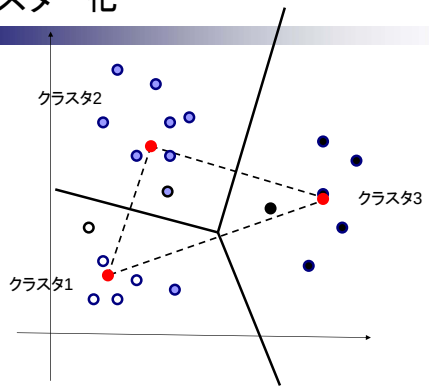
```
> iris.hc = hclust(dist(iris[,1:4]), method="ward")
> plclust(iris.hc)
> cutree(iris.hc,3)
> table(cutree(iris.hc,3),iris$Species)

> plclust(hclust(dist(iris[,1:4]), method="single"))
> plclust(hclust(dist(iris[,1:4]), method="complete"))
> plclust(hclust(dist(iris[,1:4]), method="average"))
> plclust(hclust(dist(iris[,1:4]), method="centroid"))
> plclust(hclust(dist(iris[,1:4]), method="median"))
```

非階層的クラスタ分析

- K-means法
  1. クラスタ数(=K)に最初に決定
  2. 乱数の初期値によりK個のデータ点を選択
  3. 選ばれたK個のデータ点に基づき、クラスタ境界を決定する(図4.3)
  4. 新しいクラスタ重心の計算(図4.4)
  5. 新しいクラスタ重心に基づきクラスタの再構成
  6. クラスタが固定されるまで4.-5.の繰り返し

クラスター化



Rによる非階層型クラスター分析

```
> iris.km=kmeans(dist(iris[,1:4]),3)
> names(iris.km)
[1] "cluster"  "centers"  "withinss" "size"
> iris.km$cluster
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
  2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
 19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
 37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
  2   2   2   2   2   2   2   2   2   2   2   2   2   2   3   3   3   3
 55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
  3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
 73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  3   3   3   3   3   1   3   3   3   3   3   3   3   3   3   3   3   3
 91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
  3   3   3   3   3   3   3   3   3   3   1   3   1   1   1   1   3   1
109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
  1   1   1   1   1   3   1   1   1   1   1   3   1   3   1   3   1   1
127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
  3   3   1   1   1   1   3   1   1   1   1   3   1   1   1   3   1
145 146 147 148 149 150
  1   1   3   1   1   3
```

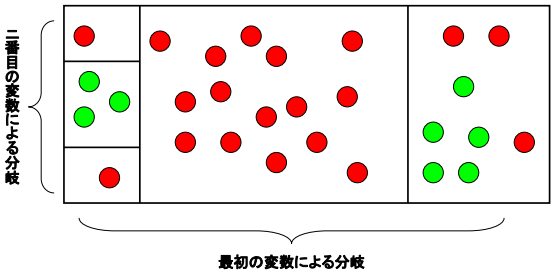

決定木

- 目的
  - 特定の結果をもたらす可能性の高いセグメントを見つける
  - 特定の結果をもたらす要因・ルールを見つける
- 利点
  - 目的変数に対する予測変数の非線形な影響を拾い出せる
  - ルールが明示される(＝解釈しやすい)
- 重要なアウトプット
  - ツリー図
- 注意点
  - 最適な解が得られるとは限らない
    上流の分岐の基準を変更すると異なったツリーが作成されることもある
    ツールを変えると異なったツリーが作成されることもある

決定木

- 決定木の仕組み
  - 説明変数を用いて目的変数を決定する手段
    - 説明変数が決定木の分岐点となる
  - 目的変数
    質的変数⇒分類木、量的変数⇒回帰木
- 木の成長
  - どの説明変数から適用するか
- 分岐の数
  - 常に2分岐 or 適宜3分岐以上

決定木出力の仕組みー分類木



回帰木

- 目的変数
  - 量的
- 説明変数
  - 質的・量的(手法により制限あり)
- 解析のポイント
  - 上位ノードが重要な変数
  - 残差により予測の良さを評価

Rによる回帰木

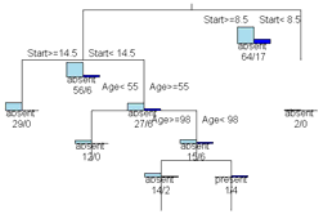
```
> data(car.test.frame,package="rpart")
> cartest.rpart=rpart(Price~.,data=car.test.frame)
> print(cartest.rpart)
n= 60
node), split, n, deviance, yval
      * denotes terminal node
 1) root 60 983551500.0 12615.670
   2) Weight< 2980 36 283686500.0 10442.580
     4) Type=Small 13  21804710.0  7682.385
       8) Mileage>=27 10   9080134.0  7160.500 *
       9) Mileage< 27 3    922158.0  9422.000 *
     5) Type=Compact,Medium,Sporty 23 106857900.0 12002.700
      10) Country=Japan,Japan/USA,Korea,USA 21  38281730.0 11487.240
        20) HP< 139 18  26603720.0 11196.830 *
        21) HP>=139 3   1051811.0 13229.670 *
      11) Country=France,Germany 2   4410450.0 17415.000 *
   3) Weight>=2980 24 274858800.0 15875.290
     6) Country=USA 14  47843040.0 14185.710
      12) Type=Compact,Sporty 3    743040.7 11551.330 *
      13) Type=Large,Medium,Van 11  20601960.0 14904.180
        26) HP< 146 5   4628639.0 13786.400 *
        27) HP>=146 6   4520155.0 15835.670 *
     7) Country=Japan,Sweden 10 131098800.0 18240.700
      14) Type=Compact,Van 6  17260570.0 15825.000
        28) Mileage< 20.5 4    677768.8 14655.250 *
        29) Mileage>=20.5 2    163020.5 18164.500 *
      15) Type=Medium 4  26304090.0 21864.250 *
> plot(cartest.rpart,uniform=T,margin=0.05)
> text(cartest.rpart,all=T,use.n=T)
```


分類木

- 目的変数
  - 質的
- 説明変数
  - 質的・量的(手法により制限あり)
- 解析のポイント
  - 上位ノードが重要な変数
  - 誤判別率で分類の正しさを評価
  - 発見されたルールをSQLなどで利用し易い

Rによる分類木

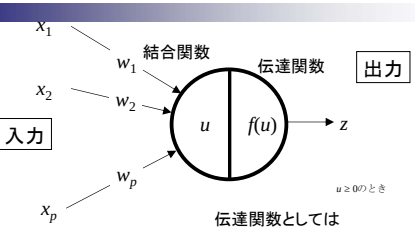
```
> library(mvpart)
> kyophosis.rpart=rpart(Kyphosis~.,data=kyphosis)
> plot(kyphosis.rpart,uniform=T,margin=0.05)
> text(kyophosis.rpart,all=T,use.n=T)
```



ニューラルネットワーク手法の分類

- 目的変数がある場合(supervised learning)
  - 目的変数=外的基準=教師信号
  - □ 階層型ネットワーク(パーセプトロン Rosenblatt 1958)
    ＋バックプロパゲーション
  - □ ボルツマンマシン (Hinton & Sejnowski 1983)
  - □ ヘルムホルツマシン (Dayan, Hinton & Radford 1995)
- 目的変数がない場合(unsupervised learning)
  - □ 自己組織化マップ
    - Kohonen(1995)
  - □ ART(Adaptive Resonance Theory)ネットワーク
    - Carpenter & Grossberg (1990)
    - Levine & Penz (1990)

単純パーセプトロン



結合関数としては

$$u = w_0 + \sum_{i=1}^{p} w_i x_i$$

伝達関数としては

$$f(u) = \begin{cases} 1 & u \geq 0 \text{のとき} \\ 0 & u < 0 \text{のとき} \end{cases}$$

$$f(u) = u$$

$$f(u) = \frac{1}{1+e^{-u}}$$

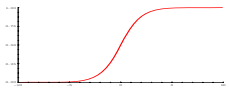
伝達関数について

$$f(u) = u \qquad z = w_0 + \sum_{i=1}^{p} w_i x_i$$

線形回帰

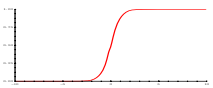
シグモイド関数

$$f(u) = \frac{1}{1+e^{-u}}$$



ロジスティック回帰

$$f(u) = \frac{1}{2}\tanh u + \frac{1}{2}$$



学習（最適な重みの推定）

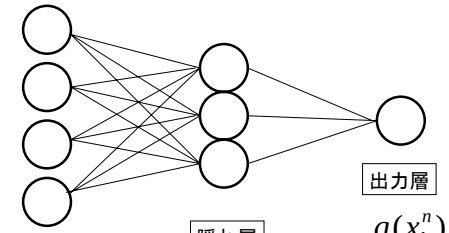
- 単純パーセプトロンの学習はデルタルール

$$R = \sum_{i=1}^{n} (z_i^* - z_i)^2 = \sum_{i=1}^{n} \delta_i^2$$

このRを最小にするようなパラメータ w_i を変化させる

$$\Delta w_k = \varepsilon (z_i^* - z_i) x_k = \varepsilon \delta x_k$$

階層型ニューラルネットワーク



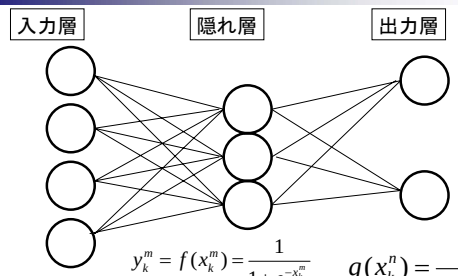
$$g(x_k^n) = x_k^n$$

$$y_k^m = f(x_k^m) = \frac{1}{1+e^{-x_k^m}}$$

学習

- バックプロパゲーション（誤差逆伝播法）
  - □ 一般化デルタルール
    - 慣性パラメータ(momentum)
    - 学習率(learning rate)
- 遺伝アルゴリズムによる学習
  - □ 大域的な最適解を見つける可能性が高い

判別が目的の場合

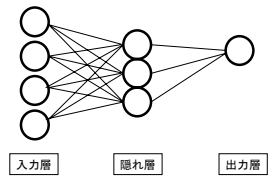


$$y_k^m = f(x_k^m) = \frac{1}{1+e^{-x_k^m}} \qquad g(x_k^n) = \frac{e^{x_k^n}}{\sum_{i=1}^{h} e^{x_i^n}}$$

ソフトマックス関数

ネットワークの設定

- 入力層の設定
  - □ 独立変数の指定
- 出力層の設定
  - □ 従属変数の指定
- 隠れ層の設定
  - □ 隠れ層の数の指定
  - □ 各隠れ層のユニット数の指定



Rによるニューラルネット

```
> library(nnet)
> kyphosis.nnet = nnet(Kyphosis~.,size=3,decay=0.1,data=kyphosis)
> summary(kyphosis.nnet)
> predict(kyphosis.nnet,type="class")
> table(kyphosis$Kyphosis,predict(kyphosis.nnet,type="class"))

         absent present
  absent      59       5
  present      6      11
#減衰重みを変更して実行
> kyphosis.nnet1=nnet(Kyphosis~.,size=3,decay=0.001,data=kyphosis)
> table(kyphosis$Kyphosis,predict(kyphosis.nnet1,type="class"))

         absent present
  absent      59       5
  present      5      12
```

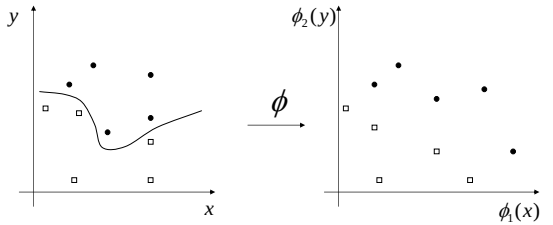

SVM (Support Vector Machine)

- データを高次元空間に埋め込んで、そこで線形分離を行う

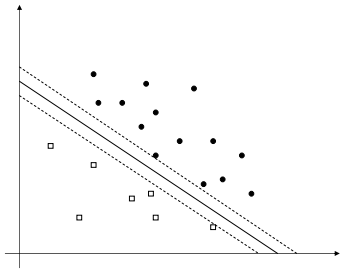
$$y = sign\left\{w_1\varphi_1(x_1,\cdots,x_p)+\cdots+w_m\varphi_m(x_1,\cdots,x_p)+b\right\}$$

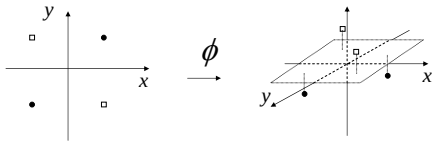
$$sign(a)=\begin{cases}1 & a\geq 0\\ -1 & a<0\end{cases}$$

一般に、mはpよりかなり大きな数。
主成分回帰と全く逆の発想



SVM





RによるSVM（回帰）

```
> library(kernlab)
> kyphsis.svm=ksvm(Kyphosis~.,data=kyphosis,kernel="rbfdot",
   kpar=list(sigma=1))
> table(kyphosis$Kyphosis,predict(kyphsis.svm))
          absent present
  absent      64       0
  present      8       9
> kyphsis.svm=ksvm(Kyphosis~.,data=kyphosis,kernel="rbfdot",
   kpar=list(sigma=5))
> table(kyphosis$Kyphosis,predict(kyphsis.svm))
> kyphsis.svm=ksvm(Kyphosis~.,data=kyphosis,kernel="rbfdot",
   kpar=list(sigma=25))
> table(kyphosis$Kyphosis,predict(kyphsis.svm))
          absent present
  absent      64       0
  present      0      17
```


MBR（k-nn）・・記憶ベース推論

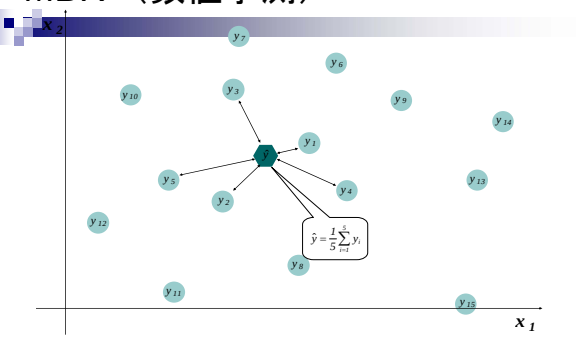
- 蓄積されたデータ(Memory)の中で、予測(または分類)すべき対象と似たものを探し出し、それらの情報のみを利用して予測(または分類)を行う

- Memory based vs Case Based
  - Memory basedの手法は、大規模データにはあまり向かないという主張がある
    - 似たものを探し出すことの困難性

K-nn の予測手順

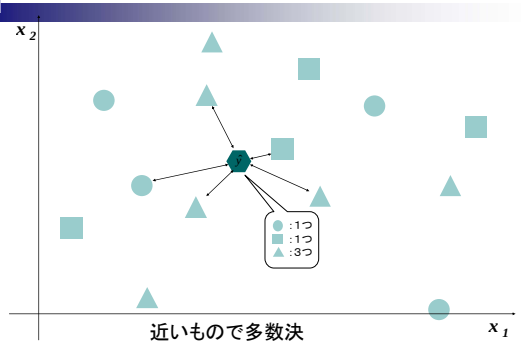
- 説明変数の尺度を変換する（量的のみ）
  - 0-1の範囲に基準化
  - 各説明変数を標準化
- 予測データと学習用データの距離の算出
  - ユークリッド距離(L_2)
  - シティブロック距離(L_1)
  - 質的変数は値が異なれば1、同一なら0
- 距離の近いk個の学習用データから予測値を求める
  - 量的目的変数・・平均
  - 質的目的変数・・最も多いカテゴリー

MBR （数値予測）



近いものでの(重みつき)平均　距離の定義 = 説明変数の選択

カテゴリー予測



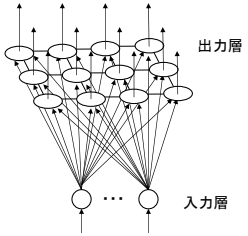
最適なk

- 記憶ベース推論では
  - kを大きくすると　予測値の分散は小さくなる
  - 予測値のバイアスが大きくなることもある
- 最適なkの値
  - クロスバリデーションにより決定

自己組織化マップ(SOM)

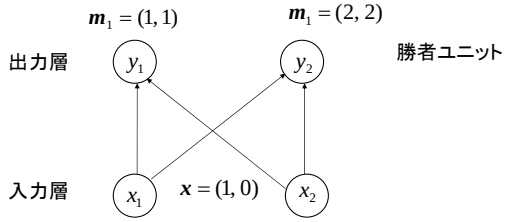
- Kohonen マップ
  - 多数の出力層を用意する
  - 各ユニットの重みだけでなく近傍のユニットの重みも調整する学習
  - 近傍のユニットとの近さが学習される

SOMの学習



単純なSOMによるネットワークの理解

$\| \boldsymbol{x}\text{-}\boldsymbol{m}_1 \| = \sqrt{(1-1)^2+(0-1)^2} = 1 \quad \| \boldsymbol{x}\text{-}\boldsymbol{m}_2 \| = \sqrt{(1-2)^2+(0-2)^2} = \sqrt{5}$

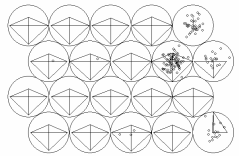


状態ベクトルの更新　$\boldsymbol{m}_1 \leftarrow \boldsymbol{m}_1 + \alpha \times h \times (\boldsymbol{x}_i - \boldsymbol{m}_k)$

重みベクトル　$h(\| \boldsymbol{r}_k - \boldsymbol{r}_j \|, t) = \exp\left(\frac{\| \boldsymbol{r}_k - \boldsymbol{r}_j \|^2}{2\sigma^2(t)} \right) \qquad \alpha(t) = 1 - \frac{t}{T}$

RによるSOM

```
> library(class)
> som.gr<-somgrid(topo="hexagonal",xdim=5,ydim=4)
> iris.som<-SOM(iris[,1:4],som.gr)
> iris.som
> plot(iris.som)
> symbols(iris.som$grid$pts[,1:2],circles=c(rep(0.5,20)),
inches=FALSE,add=TRUE)
> haizoku<-as.numeric(knn(iris.som$codes,iris[,1:4],1:20))
> rand=cbind(rnorm(nrow(iris),0,0.15),rnorm(nrow(iris),0,0.15))
> pp.new<-iris.som$grid$pts[haizoku,]+rand
> points(pp.new)
```



参考URL

- http://stat.sm.u-tokai.ac.jp/~yama/R/
- JIN’S PAGE
  - http://www1.doshisha.ac.jp/~mjin/R/index.html
- RjpWiki
  - http://www.okada.jp.org/RWiki/

おまけ

- 作業環境について
  - 作業ごとにフォルダを使い分ける
    - 共同作業では .Rdata ファイルを共有する
- 起動オプション
  - 表示言語　LANG=C
  - --SDI

参考文献

- R、S-PLUS
  - S-PLUSによるデータマイニング入門 (2005), 水田正弘 他, 森北出版
  - S-PLUSによる統計解析 (2001), Venables and Ripley (邦訳 伊藤幹夫 他), シュプリンガー・フェアラーク東京
  - Sと統計モデル—データ科学の新しい波 (1994), Chambers and Hastie (邦訳 柴田里程), 共立出版
  - データによるプログラミング—データ解析言語Sにおける新しいプログラミング (2002), John M. Chambers (著), 垂水共之(翻訳), 森北出版
  - はじめてのS-PLUS/R言語プログラミング—例題で学ぶS-PLUS/R言語の基本 (2005), 竹内俊彦, オーム社
  - A Handbook of Statistical Analysis using R (2006), Brians S. Everitt, Torsten Hothorn, Chapman & Hall/CRC
  - Using R for Introductory Statistics (2005), John Verzani, Chapman & Hall/CRC
  - 心理統計学の基礎—統合的理解のために (2002), 南風原朝和, 有斐閣

参考文献

- データマイニング全般
  - 山鳥忠司・古本　孝『戦略経営に活かすデータマイニング』2001 かんき出版
  - SASインスティチュートジャパン『データマイニングがマーケティングを変える』2001 PHPビジネス選書
- データマイニング手法
  - 豊田秀樹『金鉱を掘り当てる統計学 データマイニング入門』2001 講談社
  - 内田　治『データマイニング入門』2002 日本経済新聞社
  - マイケル・J.A. ベリー『データマイニング手法―営業、マーケティング、カスタマーサポートのための顧客分析』1999 海文堂出版
  - 福田剛志　他『データマイニング』2001 共立出版
  - SASインスティチュートジャパン共訳『データマイニング手法』1999海文堂（第2版）
  - 大滝　厚, 堀江宥治, D. Steinberg 『応用2進木解析法－CARTによる－ 』1999 サイエンス社.
  - Joseph Bigus(1997) ニューラルネットワークによるデータマイニング、日経BP社
  - 甘利俊一(1993) ニューラルネットの新展開,サイエンス社
  - 渡辺澄夫(2001) データ学習アルゴリズム,共立出版

参考文献

- 多変量統計解析
  - 山口和範, 高橋淳一, 竹内光悦 (2004)『図解入門 よくわかる多変量解析の基本と仕組み』, 秀和システム
  - 多変量統計解析法 (1983), 田中　豊・脇本和昌, 現代数学社
  - 朝野熙彦『入門 多変量解析の実際 第2版』2000 講談社サイエンティフィク
  - 柳井晴夫『多変量データ解析法-理論と応用 行動計量学シリーズ〈8〉』1994 朝倉書店
  - 竹内啓、前川眞一『SASによる多変量データの解析』東京大学出版
  - 丹後　他『ロジスティック回帰分析―SASを利用した統計解析の実際 』1996　朝倉書店
- 統計解析の注意
  - 永田靖『統計的方法のしくみ』1996 日科技連
  - 繁桝 算男 他『Q&Aで知る統計データ解析―DOs and DON'Ts』1999 サイエンス社