専門演習B（村尾）
2014/08/07

アソシエーション分析

国際文化学部3回
1226582c　田中 光

アソシエーション分析とは

- 百貨店や店舗で集めているトランザクションデータを活用するために、バスケットの中の商品間の関について分析を行う方法である

表1：買い物バスケットの事例（TID：Transaction ID）

TID	アイテム集合
1	{パン,牛乳,ハム,果物}
2	{パン,オムツ,ビール,ハム}
3	{ソーセージ,ビール,オムツ}
4	{弁当,ビール,オムツ,タバコ}
5	{弁当,ビール,ジュース,果物}

目的

- Aのようなデータから頻出するアイテムの組み合わせの規則を漏れなく抽出し、その中から興味深い結果を探し出すことを主な目的とする

アソシエーション分析における2つのアルゴリズム

①相関ルールを抽出するもの

②頻出アイテムセットを抽出するもの

①相関ルール

- トランザクションデータベースに頻出するアイテム間の何らかの組み合わせの規則

「商品Aを買うと商品Bも買う」：{A}⇒{B}　と表す

  - 左辺：条件部　　右辺：結論部

- アルゴリズム：Apriori

①-2　評価指標

- 支持度
  $supp(X \Rightarrow Y) = \sigma(X \cup Y)/M$　(Mはトランザクションの数)

- 確信度
  $\mathrm{conf}(X \Rightarrow Y) = \sigma(X \cup Y)/\sigma(X) = supp(X \Rightarrow Y)/supp(X)$

- リフト
  $lift(X \Rightarrow Y) = \mathrm{conf}(X \Rightarrow Y)/\blacksquare \mathrm{supp}(Y)$

支持度(supp)

- ルールX⇒Yの支持度：アイテム集合XとYを含むトランザクションσ（X∪Y)が全体の中に占める比率
- アイテム集合Xを含むトランザクションの数を「Xの支持度数」と呼び、「σ(X)」と表す

表１でいうところの…

X＝{オムツ,ビール}

支持度数：σ(X)＝ 3

確信度(conf) / リフト(lift)

- 確信度
  アイテム集合XとYを含むトランザクションの数
  σ（X∪Y）を、条件Xを含むトランザクションの数σ(X)で割った値である

- リフト
  確信度をsupp(Y)で割った値で定義する。
  確率Pr(X∪Y)/Pr(X)Pr(Y)　の近似値である。

支持度が高いほど、そのルールが現れる比率が低い。
しかし、アイテム数が多い大きいデータベースの
中では個別のアイテムの支持度が非常に高いことは
期待できない。

ルールの評価は、支持度、確信度、リフトを総合的に考慮する必要がある。

①-3 関数とケーススタディ

- パッケージ：arules

- 扱う形式

data.frame,tidLists,dgCMatrix,itemMatrix,
transaction,matrix形式を直接・間接に扱う
⇒関数asを用いて変換することができる

（全体図　p279：図17.1参照）

2種類のデータ表記

(a)

TID	アイテム
1	{a,b,c,d}
2	{a,e,f,g}
3	{e,f,g}
4	{e,f,h,i}
5	{d,f,h,j}

(b)

TID	a	b	c	d	e	f	g	h	i	J
1	1	1	1	1	0	0	0	0	0	0
2	1	0	0	0	1	1	1	0	0	0
3	0	0	0	0	1	1	1	0	0	0
4	0	0	0	0	1	1	0	1	1	0
5	0	0	0	1	0	1	0	1	0	1

(c)

アイテム	a	b	c	d	e	f	g	h	i	j
TID	1,2	1	1	1.5	2,3,4	2,3,4,5	2,3	4,5	4	5

表(a)形式での入力例

- 関数asを用いてtransaction形式に変換

表1のデータをすべて打ち込む（下記は一部）

```
> data<-list(c("パン","牛乳","ハム","果物"))
> data.tran<-as(data,"transactions")
> class(data.tran)
[1] "transactions"
attr(,"package")
[1] "arules"
> 
```

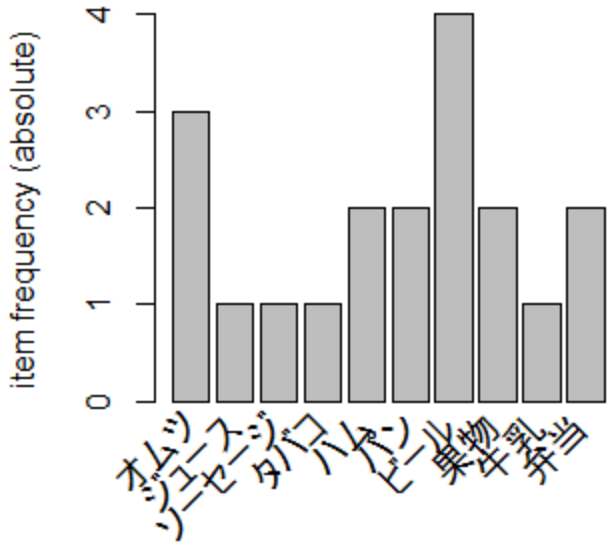
- transaction,itemMatrix形式のデータは、
  次のようにmatrix,data.frame形式に変換できる

```
> as(data.tran,"matrix")
  オムツ ジュース ソーセージ タバコ ハム パン ビール 果物 牛乳 弁当
1      0        0          0      0    1    1      0    1    1    0
2      1        0          0      0    1    1      1    0    0    0
3      1        0          1      0    0    0      1    0    0    0
4      1        0          0      1    0    0      1    0    0    1
5      0        1          0      0    0    0      1    1    0    1
> as(data.tran,"data.frame")
                          items
1         {ハム,パン,果物,牛乳}
2    {オムツ,ハム,パン,ビール}
3  {オムツ,ソーセージ,ビール}
4 {オムツ,タバコ,ビール,弁当}
5 {ジュース,ビール,果物,弁当}
>
```

データの外観を把握する

- 関数itemFrequencyを用いる
- トランザクションデータのアイテムの頻度の棒グラフを作成

```
> itemFrequencyPlot(data.tran,type="absolute")
>
```



関数apriori

- 関数aprioriを用いて相関ルールを抽出することもできる

書式：

apriori(data,parameter=NULL,appearance=NULL,control=NULL)

例1

- 入力したデータdata.tranを用いた、デフォルトのままの使用例

```
> data.ap<-apriori(data.tran)

parameter specification:
 confidence minval smax arem  aval originalSupport support minlen maxlen target   ext
        0.8    0.1    1 none FALSE            TRUE     0.1      1     10  rules FALSE

algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

 apriori(data.tran) 中で警告がありました:
  You chose a very low absolute support count of 0. You might run out of memory! Increase minimum support.

apriori - find association rules with the apriori algorithm
version 4.21 (2004.05.09)        (c) 1996-2004   Christian Borgelt
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[10 item(s), 5 transaction(s)] done [0.00s].
sorting and recoding items ... [10 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [70 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

- わずか

・5つのトランザクション

・10このアイテム

のデータセットですら70にも上るルールが抽出されている

⇒詳細に読み込むのは骨が折れる

支持度を基準としてソートした上位6位のルールを呼び出してみる

```
> inspect(head(SORT(data.ap,by="support"),n=6))
  lhs                rhs       support confidence     lift
1 {}              => {ビール}      0.8        0.8 1.000000
2 {オムツ}        => {ビール}      0.6        1.0 1.250000
3 {パン}          => {ハム}        0.4        1.0 2.500000
4 {ハム}          => {パン}        0.4        1.0 2.500000
5 {弁当}          => {ビール}      0.4        1.0 1.250000
6 {ソーセージ} => {オムツ}       0.2        1.0 1.666667
 警告メッセージ:
In .local(x, ...) : arules: SORT is deprecated use sort instead.
```

支持度が最も高い
アイテム：{ビール}
相関ルール：{オムツ}⇒{ビール}　　だと分かる。

引数prameterを指定しなおすことで、ルールの数をコントロール

- 下記では54このルールが抽出されている
- maxlen=3だから、
  抽出された1つのルールの中のアイテム数は
  最大3である（下記中略）

```
> inspect(head(SORT(data.ap1,by="support"),n=20))
   lhs                rhs       support confidence     lift
1  {}              => {ビール}     0.8         0.8 1.000000
2  {オムツ}        => {ビール}     0.6         1.0 1.250000
3  {パン}          => {ハム}       0.4         1.0 2.500000
4  {ハム}          => {パン}       0.4         1.0 2.500000
5  {弁当}          => {ビール}     0.4         1.0 1.250000
   16 {ジュース}    => {ビール}     0.2         1.0 1.250000
   17 {オムツ,
       ソーセージ} => {ビール}     0.2         1.0 1.250000
   18 {ソーセージ,
       ビール}     => {オムツ}     0.2         1.0 1.666667
   19 {パン,
       牛乳}       => {果物}       0.2         1.0 2.500000
   20 {果物,
       牛乳}       => {パン}       0.2         1.0 2.500000
```

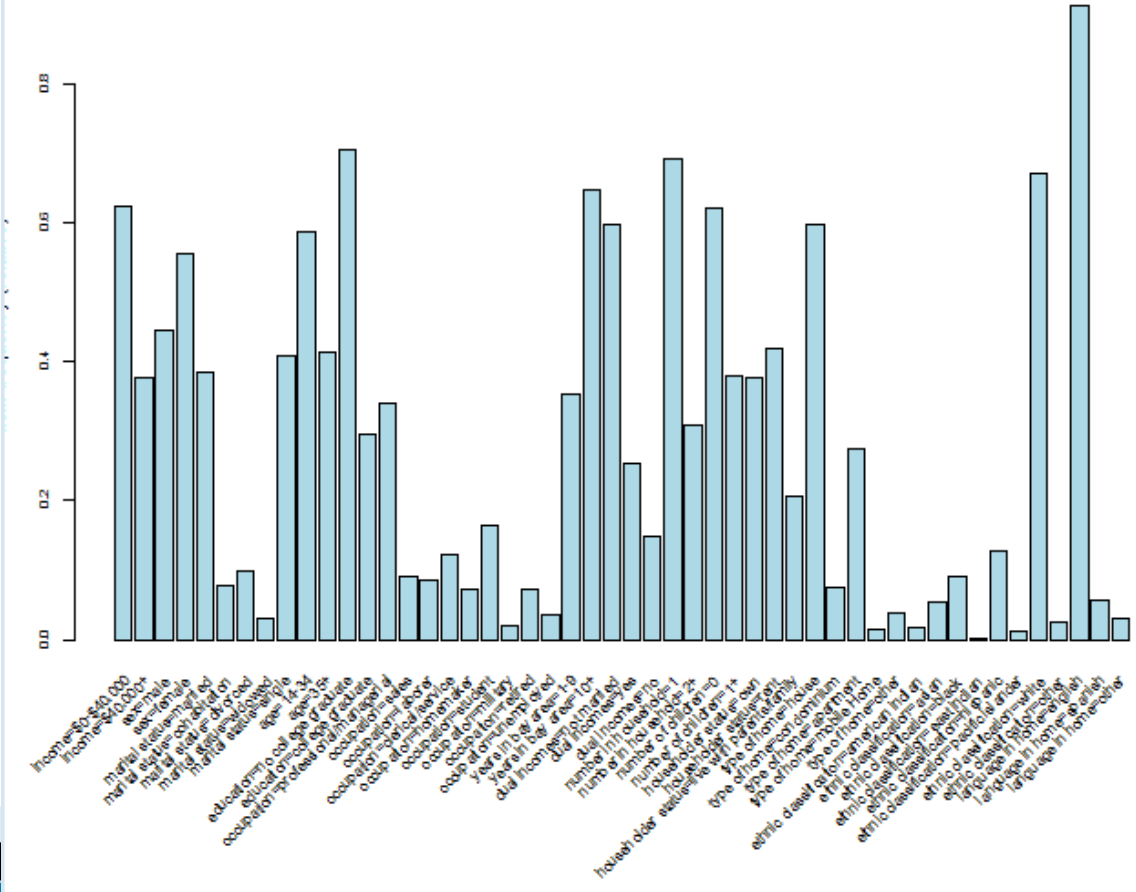
例2

- パッケージaruleにはいくつかのリアルのデータセットが用意されている

- データセット：Income
- サンフランシスコベイエリアのショッピングモールの顧客9409人が回答したアンケート14項目のデータを整理したもの

```
> data(Income)
> Income
transactions in sparse format with
 6876 transactions (rows) and
 50 items (columns)
> par(mar=c(4.5,2,1,2),cex=0.65,cex.axis=0.7)
> itemFrequencyPlot(Income,cex=0.8,col="lightblue")
```

データIncomeの
変数の相対頻度



- 関数aprioriのデフォルト値でルール抽出を行うと抽出されるルールの数は6346に上る

- 関数summary
  ルールの総数、アイテムの数ごとのルール数、評価指標の基本統計を返す

```
> Income.ap<-apriori(Income)

                中略

writing ... [8664 rule(s)] done [0.01s].
creating S4 object  ... done [0.02s].
```

```
> summary(Income.ap)
set of 8664 rules

rule length distribution (lhs + rhs):sizes
   1    2    3    4    5    6    7    8
   1   56  615 2287 3387 1925  385    8

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   4.000   5.000   4.888   6.000   8.000

summary of quality measures:
    support         confidence          lift
 Min.   :0.1001   Min.   :0.8000   Min.   :0.8971
 1st Qu.:0.1101   1st Qu.:0.8436   1st Qu.:1.0813
 Median :0.1241   Median :0.9027   Median :1.3297
 Mean   :0.1393   Mean   :0.9021   Mean   :1.4099
 3rd Qu.:0.1510   3rd Qu.:0.9574   3rd Qu.:1.5309
 Max.   :0.9129   Max.   :1.0000   Max.   :4.3554

mining info:
   data ntransactions support confidence
 Income          6876     0.1        0.8
```

- 抽出したルールを呼び出す方法は様々

・支持度、確信度、リフトの値をソート

・条件部、結論部に制約を加える

・条件部、結論部に制約を加える

- 例：結論部が高収入(income=$40,000+)かつリフトが2以上であるルールを呼び出す

```
> Income.ap2<-subset(Income.ap,subset=rhs%in%"income=$40,000+"&lift>2)
> inspect(SORT(Income.ap2)[1:3])
  lhs                                      rhs                 support confidence     lift
1 {occupation=professional/managerial,
   householder status=own}              => {income=$40,000+} 0.1384526  0.8074640 2.138722
2 {occupation=professional/managerial,
   householder status=own,
   language in home=english}            => {income=$40,000+} 0.1336533  0.8075571 2.138969
3 {dual incomes=yes,
   householder status=own}              => {income=$40,000+} 0.1260908  0.8156162 2.160315
```

・例：条件部に学歴（education）、結果部に収入(income)に関するアイテムを含むルールを呼び出す

※lsh%pi%”education”は条件部にeducationを含むことを意味する

```
> Income.ap3<-subset(Income.ap,subset=lhs%pin%"education"&rhs%pin%"income")
> inspect(SORT(Income.ap3,by="lift")[1:2])
  lhs                                  rhs                  support confidence     lift
1 {education=college graduate,
   householder status=own,
   ethnic classification=white,
   language in home=english}     => {income=$40,000+} 0.1002036  0.8115430 2.149526
2 {education=college graduate,
   householder status=own,
   type of home=house,
   language in home=english}     => {income=$40,000+} 0.1031123  0.8112128 2.148652
```

例3

- トランザクションデータセット：Groceries
  ローカルの食料雑貨店のPOSデータ

- カテゴリ数が多いので(169)、
  itemFrequencyを用いる場合は図を分ける

```
> data(Groceries)
> par(mfrow=c(1,3),mar=c(4.5,2,1,2),cex=0.65,cex.axis=0.7)
> itemFrequencyPlot(Groceries[,1:55],cex=0.65,col="lightblue",horiz=T)
> itemFrequencyPlot(Groceries[,56:110],cex=0.65,col="lightblue",horiz=T)
> itemFrequencyPlot(Groceries[,111:169],cex=0.65,col="lightblue",horiz=T)
```

- トランザクションの中のアイテム数が多く、分散が大きいときには、高いsupport値を持つルールが得難い。

⇒デフォルトよりパラメータの値を下げて関数aprioriを実行する

```
> Gr.ap<-apriori(Groceries,parameter=list(support=0.005,confidence=0.01))

parameter specification:
 confidence minval smax arem  aval originalSupport support minlen maxlen target   ext
       0.01    0.1    1 none FALSE            TRUE   0.005      1     10  rules FALSE

algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

apriori - find association rules with the apriori algorithm
version 4.21 (2004.05.09)        (c) 1996-2004   Christian Borgelt
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
sorting and recoding items ... [120 item(s)] done [0.00s].
creating transaction tree ... done [0.01s].
checking subsets of size 1 2 3 4 done [0.02s].
writing ... [2138 rule(s)] done [0.00s].
creating S4 object  ... done [0.01s].
```

②-1頻出アイテムの抽出

- 今回は深さ優先検索のアルゴリズム：Eclat
  を用いる

- その結果は{X⇒Y}ではなく、
  頻出するアイテムの組み合わせの集合{X,Y}

書式：eclat(data,parameter=NULL,control=NULL)

例4

- 表1に示すデータを用いた例を示す

```
> data.ec<-eclat(data.tran)
```

```
> inspect(SORT(data.ec,by="support")[1:5])
  items     support
1 {ビール}      0.8
2 {オムツ,
   ビール}      0.6
3 {オムツ}      0.6
4 {ビール,
   弁当}        0.4
5 {ハム,
   パン}        0.4
```

注目すべき点

- 1つのアイテムの結果も呼び出されている

⇒組み合わせに注目する際にこのでようなアイテムは必要ない。

⇒関数sizeを用いてアイテム数を指定して呼び出す

```
> data.ec2<-data.ac[size(items(data.ec))==2]
> inspect(SORT(data.ec2,by="support")[1:5])
  items                 support
1 {オムツ,
   ビール}                0.6
2 {ビール,
   弁当}                  0.4
3 {ハム,
   パン}                  0.4
4 {ソーセージ,
   ビール}                0.2
5 {オムツ,
   ソーセージ}            0.2
```

- 支持度が最も高いアイテムの組み合わせは |オムツ,ビール|である

⇒関数aprioriを用いた結果と同じ

例5

- Incomeを用いた場合も、関数aprioriと同じく制約条件を付けることで頻出アイテムセットを呼び出すことができる。

- ただし、eclatには条件部・結果部はない
  アイテムセットと支持度のみである

- アイテムセットの中にincome=$40,000を含み、支持度が0.2以上のものを抽出する実行例

```
> Income.ec<-eclat(Income)

> Income.ec2<-subset(Income.ec,subset=items%in%"income=$40,000+"&support>0.2&size(items)>2)
> Income.ec2
set of 17 itemsets
> inspect(SORT(Income.ec2,by="support")[1:2])
  items                                 support
1 {income=$40,000+,
   ethnic classification=white,
   language in home=english}    0.2789412
2 {income=$40,000+,
   type of home=house,
   language in home=english}    0.2613438
```

②-2 抽出結果の補助分析

- 抽出したルールや頻出アイテムについて
  クラスター分析を行い、クラスの特徴を
  考察することも有益

- 関数dissimilarity
  トランザクション、アイテム、相関ルール、
  頻出アイテムの距離（非類似度）が求められる

- 書式

  dissimilarity(x,y=NULL,method=NULL,arg=NULL,...)

- 引数method
  バイナリデータ
  (jaccard,matching,dice,affinity)を指定できる

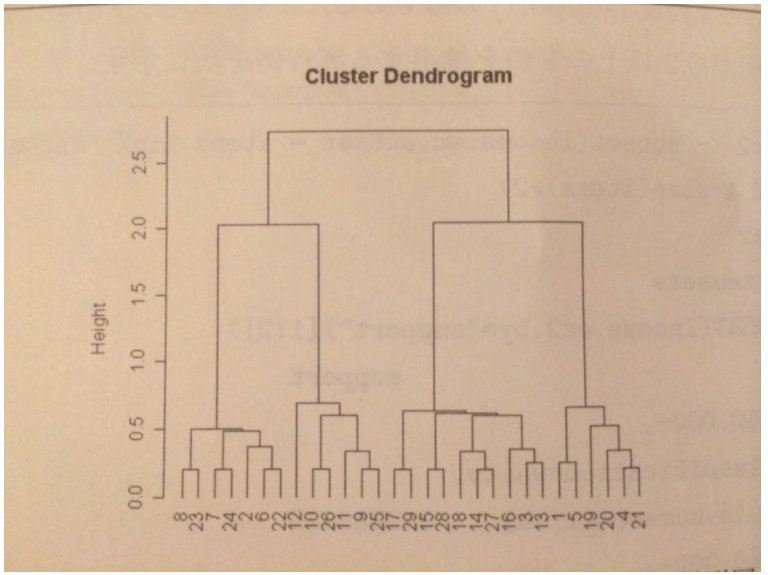
  デフォルトではjaccardが指定されている

ルールのクラスター分析

- 例2の結果部に高収入(income=$40.000+)になっている相関ルールについて、階層的クラスター分析を用いた例

```
> rules.sub<-subset(Income.ap,subset=rhs%in%"income=$40,000+"&lift>2)
> d<-dissimilarity(rules.sub)
> plot(hclust(d,"ward"),hang=-1)
```

- 樹形図から、ルールはおおまかに2つか4つのクラスに分けられる



4つのクラスに分けると…

class1=1:7　　class2=8:13

class3=14:23　　class4=24:29

葉の番号は関数hclustの結果のオブジェクトである
$orderで呼び出すことができる