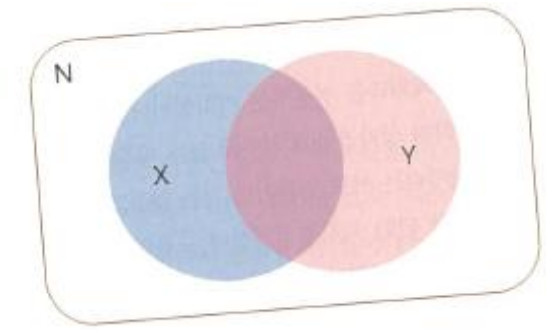
Chapter 9. Association Analysis

1. 생활 속의 문제

도시계획 관련 정책 입안: 늘 같은 건물에 있는 경향에 있는 상가 파악하기

2. 알고리즘

연관분석 (association rule): {병원}->{약국} association rule 이 생성되었다면 확률은?
연관규칙 X->Y 에서 지지도와 신뢰도는 다음과 같음.

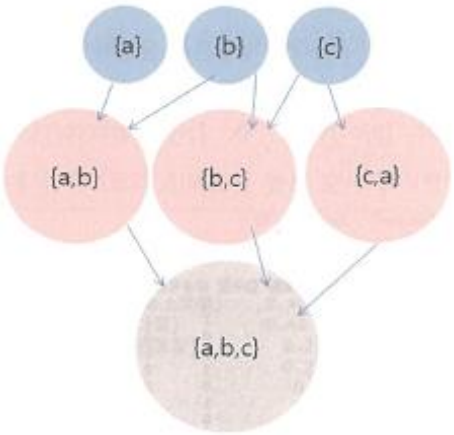


- support s(X->Y)=n(X^Y)/N: 전체 건수 중에서 X 와 Y 가 모두 포함되어 있는 건수의 비=>X 와 Y 가 동시에 존재할 확률
- confidence d(X->Y)=n(X^Y)/n(X): 항목 X 를 포함하는 건수 중에서 X 와 Y 를 모두 포함하는 건수의 비=>X 에 존재할 때에 Y 가 존재할 조건부 확률

최소 지지도를 정하여 그 중 신뢰도가 어느 정도 높은 것으로 선정한다.

빈발항목집합(frequent itemset): 최소지지도 이상을 갖는 항목집합

빈발항목집합 참고자료 참고자료 2



선험적 규칙(Apriori Principle) 모든 항목집합에 대한 지지도를 계산하지 않고 원하는 빈발항목집합을 찾아내는데 이용되는 선험적 규칙은 다음과 같다. 1) 한 항목집합이 빈발하다면, 이 항목집합의 모든 부분집합은 역시 빈발항목집합이다. 2) 한 항목집합이 비반발하다면, 이 항목집합을 포함하는 모든 집합은 비빈발항목집합이다.

연관규칙 평가 척도 : 지지도(Support), 신뢰도(Confidence), 향상도(Lift)

$$Lift(A,B) = c(A -> B)/s(B)$$

Lift=1: A,B 가 독립 Lift>1: A,B 가 양의 상관 관계 Lift<1: A,B 가 음의 상관 관계

3. 코딩과 구현

1) 데이터 전처리

```
#getwd()
build <- read.csv("data/building.csv" , header = T) #window
#build <- read.csv("data/re-encode/building.csv" , header = T) #mac

#Error in make.names(col.names, unique = TRUE) : invalid multibyte string at
'<ec><95><bd>援<ad>'

#Warning message:
#In strsplit(code, "\n", fixed = TRUE) :
#  input string 1 is invalid in this locale

build[is.na(build)] <- 0
build <- build[-1]
build
```

```
##    병원 약국 카페 휴대폰매장 일반음식점 패밀리레스토랑 당구장 보습학원
## 1     1    1    0          1          0              0      0        0
## 2     0    0    0          0          1              1      1        0
## 3     0    0    1          0          0              0      0        1
## 4     0    0    1          0          0              0      0        1
## 5     1    0    0          0          1              1      0        0
## 6     0    0    1          1          0              0      0        0
## 7     0    0    0          0          0              1      0        0
## 8     1    1    0          1          0              0      0        0
## 9     0    0    0          0          0              0      1        0
## 10    0    0    0          0          1              1      0        0
## 11    0    0    1          0          0              0      0        1
## 12    0    0    0          0          1              1      1        0
## 13    1    1    0          1          0              0      0        0
## 14    0    0    0          0          1              1      0        0
## 15    0    0    0          0          1              1      0        0
## 16    0    0    1          0          0              0      0        1
## 17    0    0    0          0          1              1      1        0
## 18    1    1    0          1          0              0      0        0
## 19    0    0    0          0          1              1      1        0
## 20    1    1    0          1          0              0      0        0
##    슈퍼마켓 은행 편의점 화장품
## 1         0    0      0      1
## 2         0    0      1      0
## 3         0    1      0      0
## 4         0    1      0      0
## 5         0    0      1      1
## 6         0    0      0      0
## 7         0    0      0      1
## 8         0    0      0      0
## 9         1    0      0      0
## 10        0    0      1      1
## 11        0    1      0      0
## 12        0    0      0      0
## 13        0    0      0      0
## 14        0    0      1      1
## 15        0    0      1      1
## 16        0    1      0      0
## 17        0    0      0      0
## 18        0    0      0      0
## 19        0    0      0      0
## 20        0    0      0      0
```

2) 모델링과 규칙 생성

참고자료: R 연관규칙 (Association Rule): R arules package 로 연관규칙 분석하기

```
#install.packages("arules")
library(arules)

## Loading required package: Matrix

##
## Attaching package: 'arules'

## The following objects are masked from 'package:base':
##
##     abbreviate, write

trans <- as.matrix(build , "Transaction")
rules1 <- apriori(trans , parameter = list(supp=0.2 , conf = 0.6 , target = "
rules"))

## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5     0.2      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 4
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[12 item(s), 20 transaction(s)] done [0.00s].
## sorting and recoding items ... [11 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [46 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].

#rules1 #Set of 46 rule(s)
```

3) 어떤 규칙이 생성되었는지 탐사. 규칙 중 지지도와 신뢰도가 높은 순서로 정렬

```
inspect(sort(rules1))

##      lhs                                rhs                support
## [1]  {일반음식점}                    => {패밀리레스토랑} 0.40
## [2]  {패밀리레스토랑}                => {일반음식점}     0.40
## [3]  {약국}                          => {휴대폰매장}     0.25
## [4]  {휴대폰매장}                    => {약국}           0.25
```

```
## [5]  {약국}                          => {병원}           0.25
## [6]  {병원}                          => {약국}           0.25
## [7]  {휴대폰매장}                    => {병원}           0.25
## [8]  {병원}                          => {휴대폰매장}     0.25
## [9]  {편의점}                        => {일반음식점}     0.25
## [10] {일반음식점}                    => {편의점}         0.25
## [11] {편의점}                        => {패밀리레스토랑} 0.25
## [12] {화장품}                        => {패밀리레스토랑} 0.25
## [13] {약국,휴대폰매장}               => {병원}           0.25
## [14] {병원,약국}                     => {휴대폰매장}     0.25
## [15] {병원,휴대폰매장}               => {약국}           0.25
## [16] {일반음식점,편의점}             => {패밀리레스토랑} 0.25
## [17] {패밀리레스토랑,편의점}         => {일반음식점}     0.25
## [18] {일반음식점,패밀리레스토랑}     => {편의점}         0.25
## [19] {보습학원}                      => {은행}           0.20
## [20] {은행}                          => {보습학원}       0.20
## [21] {보습학원}                      => {카페}           0.20
## [22] {카페}                          => {보습학원}       0.20
## [23] {은행}                          => {카페}           0.20
## [24] {카페}                          => {은행}           0.20
## [25] {당구장}                        => {일반음식점}     0.20
## [26] {당구장}                        => {패밀리레스토랑} 0.20
## [27] {편의점}                        => {화장품}         0.20
## [28] {화장품}                        => {편의점}         0.20
## [29] {화장품}                        => {일반음식점}     0.20
## [30] {보습학원,은행}                 => {카페}           0.20
## [31] {카페,보습학원}                 => {은행}           0.20
## [32] {카페,은행}                     => {보습학원}       0.20
## [33] {일반음식점,당구장}             => {패밀리레스토랑} 0.20
## [34] {패밀리레스토랑,당구장}         => {일반음식점}     0.20
## [35] {편의점,화장품}                 => {일반음식점}     0.20
## [36] {일반음식점,편의점}             => {화장품}         0.20
## [37] {일반음식점,화장품}             => {편의점}         0.20
```

```
## [38] {편의점,화장품}                       => {패밀리레스토랑} 0.20
## [39] {패밀리레스토랑,편의점}               => {화장품}         0.20
## [40] {패밀리레스토랑,화장품}               => {편의점}         0.20
## [41] {일반음식점,화장품}                   => {패밀리레스토랑} 0.20
## [42] {패밀리레스토랑,화장품}               => {일반음식점}     0.20
## [43] {일반음식점,편의점,화장품}            => {패밀리레스토랑} 0.20
## [44] {패밀리레스토랑,편의점,화장품}        => {일반음식점}     0.20
## [45] {일반음식점,패밀리레스토랑,편의점} => {화장품}         0.20
## [46] {일반음식점,패밀리레스토랑,화장품} => {편의점}         0.20
##      confidence lift
## [1]  1.0000000  2.222222
## [2]  0.8888889  2.222222
## [3]  1.0000000  3.333333
## [4]  0.8333333  3.333333
## [5]  1.0000000  3.333333
## [6]  0.8333333  3.333333
## [7]  0.8333333  2.777778
## [8]  0.8333333  2.777778
## [9]  1.0000000  2.500000
## [10] 0.6250000  2.500000
## [11] 1.0000000  2.222222
## [12] 0.8333333  1.851852
## [13] 1.0000000  3.333333
## [14] 1.0000000  3.333333
## [15] 1.0000000  4.000000
## [16] 1.0000000  2.222222
## [17] 1.0000000  2.500000
## [18] 0.6250000  2.500000
## [19] 1.0000000  5.000000
## [20] 1.0000000  5.000000
## [21] 1.0000000  4.000000
## [22] 0.8000000  4.000000
## [23] 1.0000000  4.000000
## [24] 0.8000000  4.000000
## [25] 0.8000000  2.000000
## [26] 0.8000000  1.777778
## [27] 0.8000000  2.666667
## [28] 0.6666667  2.666667
## [29] 0.6666667  1.666667
## [30] 1.0000000  4.000000
## [31] 1.0000000  5.000000
## [32] 1.0000000  5.000000
## [33] 1.0000000  2.222222
## [34] 1.0000000  2.500000
## [35] 1.0000000  2.500000
```

```
## [36] 0.8000000  2.666667
## [37] 1.0000000  4.000000
## [38] 1.0000000  2.222222
## [39] 0.8000000  2.666667
## [40] 0.8000000  3.200000
## [41] 1.0000000  2.222222
## [42] 0.8000000  2.000000
## [43] 1.0000000  2.222222
## [44] 1.0000000  2.500000
## [45] 0.8000000  2.666667
## [46] 1.0000000  4.000000
```

일반음식점과 패밀리레스토랑이 같이 있는 경향이 높고, 약국과 휴대폰 매장과 병원이 같은 건물에 있을 확률이 높다. 보습학원이 있으면 은행이나 카페가 있는 경우가 많다.

'보습학원'이 포함되어 있는 규칙만 뽑아보자.

```
rules2 <- subset(rules1 , subset = lhs %pin% '보습학원' & confidence > 0.7)
inspect(sort(rules2))
```

```
##     lhs                  rhs    support confidence lift
## [1] {보습학원}       => {은행} 0.2     1          5
## [2] {보습학원}       => {카페} 0.2     1          4
## [3] {보습학원,은행} => {카페} 0.2     1          4
## [4] {카페,보습학원} => {은행} 0.2     1          5
```

결과에 해당하는 부분에 '편의점'을 넣어보자.

```
rules3 <- subset(rules1 , subset = rhs %pin% '편의점' & confidence > 0.7)
# rules3 #Set of 3 rules
inspect(sort(rules3))
```

```
##     lhs                                  rhs      support confidence lift
## [1] {일반음식점,화장품}                 => {편의점} 0.2     1.0        4.0
## [2] {패밀리레스토랑,화장품}             => {편의점} 0.2     0.8        3.2
## [3] {일반음식점,패밀리레스토랑,화장품} => {편의점} 0.2     1.0        4.0
```

보통 음식점이나 화장품 가게 있는 곳은 높은 확률로 편의점이 들어옴.

4. 시각화

SNA(Social Network Analysis)에 사용되는 Graph 로 연관탐사 결과를 설명

```r
#visualization
b2 <- t(as.matrix(build)) %*% as.matrix(build) #상가간의 관계를 나타내는 행렬
b2
```

```
##                 병원 약국 카페 휴대폰매장 일반음식점 패밀리레스토랑 당구장
## 병원               6    5    0          5          1              1      0
## 약국               5    5    0          5          0              0      0
## 카페               0    0    5          1          0              0      0
## 휴대폰매장         5    5    1          6          0              0      0
## 일반음식점         1    0    0          0          8              8      4
## 패밀리레스토랑     1    0    0          0          8              9      4
## 당구장             0    0    0          0          4              4      5
## 보습학원           0    0    4          0          0              0      0
## 슈퍼마켓           0    0    0          0          0              0      1
## 은행               0    0    4          0          0              0      0
## 편의점             1    0    0          0          5              5      1
## 화장품             2    1    0          1          4              5      0
##                 보습학원 슈퍼마켓 은행 편의점 화장품
## 병원                   0        0    0      1      2
## 약국                   0        0    0      0      1
## 카페                   4        0    4      0      0
## 휴대폰매장             0        0    0      0      1
## 일반음식점             0        0    0      5      4
## 패밀리레스토랑         0        0    0      5      5
## 당구장                 0        1    0      1      0
## 보습학원               4        0    4      0      0
## 슈퍼마켓               0        1    0      0      0
## 은행                   4        0    4      0      0
## 편의점                 0        0    0      5      4
## 화장품                 0        0    0      4      6
```

```r
#install.packages('sna')
library(sna)
```

```
## Loading required package: statnet.common
```

```
## Loading required package: network
```

```
## network: Classes for Relational Data
## Version 1.13.0 created on 2015-08-31.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
##                     Mark S. Handcock, University of California -- Los Ange
les
##                     David R. Hunter, Penn State University
##                     Martina Morris, University of Washington
##                     Skye Bender-deMoll, University of Washington
##  For citation information, type citation("network").
##  Type help("network-package") to get started.

## sna: Tools for Social Network Analysis
## Version 2.4 created on 2016-07-23.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
##  For citation information, type citation("sna").
##  Type help(package="sna") to get started.
```

```
#source("http://bioconductor.org/biocLite.R")
#biocLite("rgl")
library(rgl)
b2.w <- b2 - diag(diag(b2))
b2.w
```

```
##                병원 약국 카페 휴대폰매장 일반음식점 패밀리레스토랑 당구장
## 병원              0    5    0          5          1              1      0
## 약국              5    0    0          5          0              0      0
## 카페              0    0    0          1          0              0      0
## 휴대폰매장        5    5    1          0          0              0      0
## 일반음식점        1    0    0          0          0              8      4
## 패밀리레스토랑    1    0    0          0          8              0      4
## 당구장            0    0    0          0          4              4      0
## 보습학원          0    0    4          0          0              0      0
## 슈퍼마켓          0    0    0          0          0              0      1
## 은행              0    0    4          0          0              0      0
## 편의점            1    0    0          0          5              5      1
## 화장품            2    1    0          1          4              5      0
##                보습학원 슈퍼마켓 은행 편의점 화장품
## 병원                  0        0    0      1      2
## 약국                  0        0    0      0      1
## 카페                  4        0    4      0      0
## 휴대폰매장            0        0    0      0      1
```

```
## 일반음식점              0        0    0      5      4
## 패밀리레스토랑          0        0    0      5      5
## 당구장                  0        1    0      1      0
## 보습학원                0        0    4      0      0
## 슈퍼마켓                0        0    0      0      0
## 은행                    4        0    0      0      0
## 편의점                  0        0    0      0      4
## 화장품                  0        0    0      4      0
```

```
#rownames(b2.w)
#colnames(b2.w)
gplot(b2.w , displaylabel=T , vertex.cex=sqrt(diag(b2)) , vertex.col = "green
" , edge.col="blue" , boxed.labels=F , arrowhead.cex = .3 , label.pos = 3 , e
dge.lwd = b2.w*2)
```