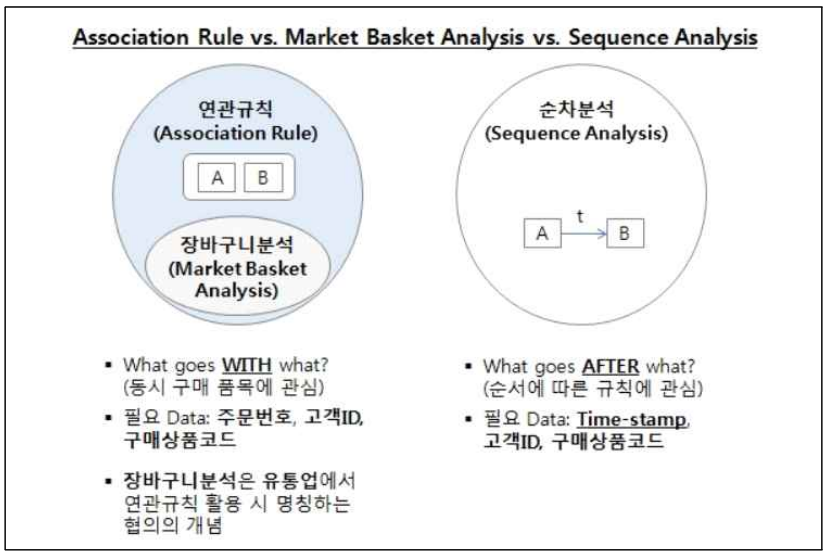
21-1 YDMS 8주차 과제

< Subject : Association Rule & Collaborative Filtering >

김하은

✓ 이론

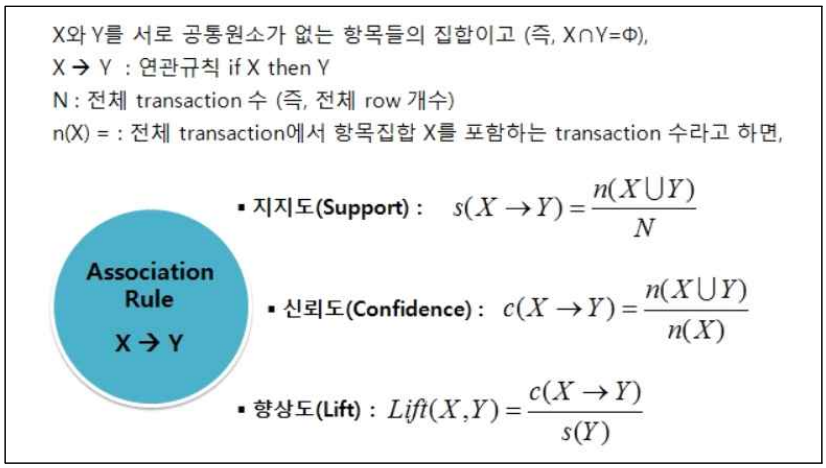
비지도학습의 하나인 연관규칙은 특정 사건이 발생하였을 때 빈번하게 발생하는 또 다른 사건의 규칙을 말한다. 여기서 규칙은 If-Then 구조로 표현을 한다. 규칙의 논리는 If-Then 구조이지만, 규칙 계산은 확률로 한다. 한편 연관규칙분석은 '동시' 구매 품목에 관심을 가지는데 비해, 순차분석은 '시간의 순서'에 따른 규칙에 관심을 가지는 분석이다. 따라서 순차분석에는 "Time Stamp" 변수가 추가로 필요하다.



연관분석에서 아이템 세트를 구성할 때 주의해야 할 점은, 아이템 세트는 구매한 조합이 아니라 가능한 아이템 조합이라는 것이다. 따라서 p개의 서로 다른 아이템 세트들이 가능한 조합은, p의 크기가 늘어날수록 지수적으로 증가한다. 막대한 크기일수록 계산에 어려움을 겪기 때문에 상대적으로 빈도수가 높은 조합만 고려하여 빈발 아이템 세트라는 새로운 아이템 세트를 구상하였다.

한편 연관규칙은 여러 척도를 통해 계산할 수 있다. 대표적으로 지지도, 신뢰도, 향상도, IS측도, 교차지지도가 있다.

- 지지도 : 지지도는 두 항목집합 X와 Y의 전체 거래 건수 중에서 X와 Y를 모두 모함하는 거래수의 비율을 말한다.
- 신뢰도 : 신뢰도는 항목집합 X를 포함하는 거래 중에서 항목집합 Y도 포함하는 거래 비율을 말한다. 신뢰도가 높을수록 유용한 규칙일 가능성이 높다.
- 향상도 : 항목집합 X가 주어지지 않았을 때의 항목집합 Y의 확률 대비 항목집합 X가 주어졌을 때, 항목집합 Y의 확률 증가 비율을 말한다. 향상도가 1보다 크거나 작다면 우수하다는 것을 의미한다.



- IS측도 : 향상도와 지지도의 곱에 제곱근을 취한 값이다. 따라서 향상도와 지지도가 모두 높을수록 IS 측도고 커진다. 둘 중 하나라도 값이 작아지면 IS 측도도 작아진다.

$$IS(A,\ B) = \sqrt{Lift(A,B) \times s(A,B)} = \frac{s(A,B)}{\sqrt{s(A),s(B)}}$$

- 교차 지지도 : 최대 지지도에 대한 최소 지지도의 비율이다. 항목집합에 대하여 의미없는 연관규칙의 생성을 방지하기 위하야 교차지지도를 이용한다. 분자에는 최소 지지도를, 분모에는 최대 지지도를 이용하여 계산하므로 이 차이가 커지면 교차지지도는 낮아지게 된다. 이 비율이 작으면 연관규칙이 의미 없을 가능성이 크다.

$$r(X) = \frac{min\{s(i_1),\ s(i_2),\ \ldots,\ s(i_m)\}}{max\{s(i_1),\ s(i_2),\ \ldots,\ s(i_m)\}}$$

다음으로 연관규칙에서 사용하는 대표적인 알고리즘으로는 Apriori algorithm 이 있다. 이 알고리즘의 핵심은 하나의 아이템만으로 이루어진 빈발 아이템 세트를 생성하면서 시작한 후 모든 크기의 빈발 아이템 세트를 생성할 때까지 재귀적으로 생성한다는 것이다. (빈발 아이템 세트란 최소 지지도 이상을 갖는 항목집합을 뜻한다.)

- 빈발항목집합 추출, Apriori 알고리즘의 원리
  1) 한 항목집합이 빈발(frequent)하다면 이 항목집합의 모든 부분집합은 역시 빈발항목집합이다. (frequent item sets -> next step)
  2) 한 항목집합이 비 빈발(infrequent)하다면 이 항목집합을 포함하는 모든 집합은 비 빈발 항목집합이다. (superset -> pruning)

협업필터링은 쉽게 말해 추천 시스템에 사용되는 기술로, 사용자들의 다양한 선호도를 고려하여 방대한 양의 항목집합으로부터 연관성이 있는 항목들을 특정 사용자에게 알려준다는 개념에 기반을 둔다. 협업필터링은 크게 ① 사용자 기반 협업 필터링, ② 항목 기반 협업 필터링이 있다.

우선 사용자 기반 협업 필터링은 피어슨 상관계수를 이용하여 사용자들 간 거리를 측정하고, 한계점을 거리 혹은 필요한 이웃들의 개수에 적용하여 최근접 이웃을 결정하는데 사용할 수 있다. 이를 사용자 기반 최우선 N 추천이라 한다. 피어슨 상관계수 말고, 코사인 유사도를 이용하여 측정할 수도 있는데, 평균을 빼지 않는다는 것이 특징이다.

다음으로 항목 기반 협업 필터링은 사용자 수가 항목들의 수보다 훨씬 큰 경우 효율적이다. 항목 기반 협업 필터링은 임의의 사용자의 관심 상품을 찾고, 비슷한 항목들 중에서 가장 대중적이거나 상관관계가 높은 항목을 추천하는 방식으로 구성된다. 이 경우 사용자가 아닌 상품의 유사도가 계산된다.

✓ 실습

< 1. prodsales, prodhierarchy 데이터를 사용하여 분석 진행 >

주어진 데이터 셋은 sas data 였으므로 sas7bdat 패키지를 이용하여 불러왔다.

	Customer	Item
1	Anne	Low Fat Milk
2	Anne	Cheddar Cheese
3	Anne	Cake
4	Anne	Frozen Pizza
5	Anne	Ice Cream
6	Anne	Pancakes
7	Bob	Low Fat Milk
8	Bob	Swiss Cheese
9	Bob	Frozen Pizza
10	Bob	Ice Cream
11	Chris	Skim Milk
12	Chris	Swiss Cheese
13	Chris	Ice Cream
14	Chris	Cake

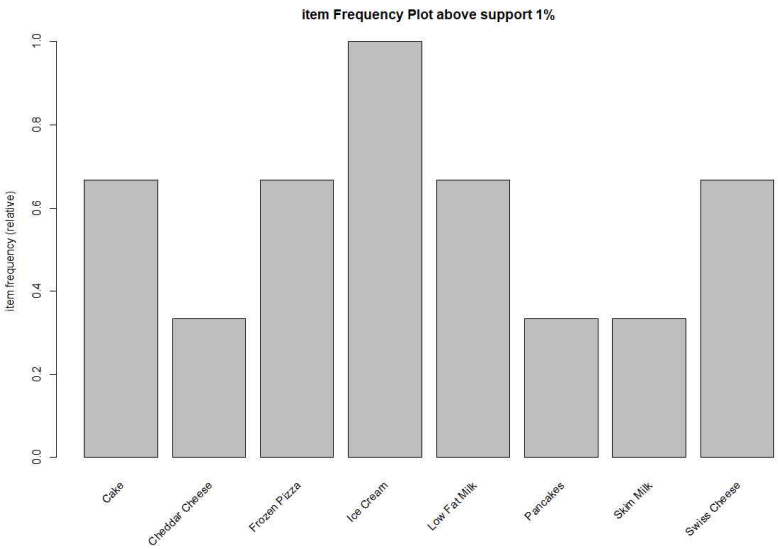
이후 arules 패키지를 이용하여 연관규칙을 분석하기 위해 데이터 형태를 변환해주었다. split 함수는 customer 별로 하나의 레코드가 되도록 변환해주는 함수이다. 또한 주어진 벡터의 요인별로 데이터를 구분하여 list를 생성해준다.

```
> ps.list<-split(prodsales$Item,prodsales$Customer)
> ps.list
$Anne
[1] Low Fat Milk   Cheddar Cheese Cake          Frozen Pizza   Ice Cream      Pancakes
Levels: Cake Cheddar Cheese Frozen Pizza Ice Cream Low Fat Milk Pancakes Skim Milk Swiss Cheese

$Bob
[1] Low Fat Milk Swiss Cheese Frozen Pizza Ice Cream
Levels: Cake Cheddar Cheese Frozen Pizza Ice Cream Low Fat Milk Pancakes Skim Milk Swiss Cheese

$Chris
[1] Skim Milk    Swiss Cheese Ice Cream    Cake
Levels: Cake Cheddar Cheese Frozen Pizza Ice Cream Low Fat Milk Pancakes Skim Milk Swiss Cheese

> ps.trans<-as(ps.list,"transactions")
> ps.trans
transactions in sparse format with
 3 transactions (rows) and
 8 items (columns)
```

위의 결과를 보면, 3명의 구매자와 8개의 물품으로 구성된 데이터임을 알 수 있다. 좀더 자세한 정보를 보기 위해 summary 함수를 이용해보았다.

```
> summary(ps.trans)
transactions as itemMatrix in sparse format with
 3 rows (elements/itemsets/transactions) and
 8 columns (items) and a density of 0.5833333

most frequent items:
   Ice Cream         Cake Frozen Pizza Low Fat Milk Swiss Cheese      (Other)
           3            2            2            2            2            3

element (itemset/transaction) length distribution:
sizes
4 6
2 1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.000   4.000   4.000   4.667   5.000   6.000

includes extended item information - examples:
          labels
1           Cake
2 Cheddar Cheese
3   Frozen Pizza

includes extended transaction information - examples:
  transactionID
1          Anne
2           Bob
3         Chris
```

위의 결과를 보면, Ice Cream이 3회로 가장 빈번하게 구매된 물품임을 알 수 있다. 다만 거래 물품 수는 2-3회로 대부분 비슷했다. 이를 시각적으로 나타낸 그래프는 아래와 같다.



```
> ps_rule<-apriori(ps.trans)
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.8    0.1    1 none FALSE            TRUE       5     0.1      1     10  rules TRUE

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

Absolute minimum support count: 0

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[8 item(s), 3 transaction(s)] done [0.00s].
sorting and recoding items ... [8 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 done [0.00s].
writing ... [179 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].

> summary(ps_rule)
set of 179 rules

rule length distribution (lhs + rhs):sizes
 1  2  3  4  5  6
 1 19 59 64 30  6

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   3.000   4.000   3.676   4.000   6.000

summary of quality measures:
    support         confidence    coverage         lift           count
 Min.   :0.3333   Min.   :1    Min.   :0.3333   Min.   :1.000   Min.   :1.000
 1st Qu.:0.3333   1st Qu.:1    1st Qu.:0.3333   1st Qu.:1.500   1st Qu.:1.000
 Median :0.3333   Median :1    Median :0.3333   Median :1.500   Median :1.000
 Mean   :0.3538   Mean   :1    Mean   :0.3538   Mean   :1.771   Mean   :1.061
 3rd Qu.:0.3333   3rd Qu.:1    3rd Qu.:0.3333   3rd Qu.:3.000   3rd Qu.:1.000
 Max.   :1.0000   Max.   :1    Max.   :1.0000   Max.   :3.000   Max.   :3.000

mining info:
     data ntransactions support confidence
 ps.trans             3     0.1        0.8
```

다음은 apriori 알고리즘을 이용한 결과이다. 최소 지지도와 신뢰도 설정 없이 진행해본 결과 179개의 규칙이 생성됨을 볼 수 있고, 최소 지지도는 0.333, 신뢰도는 1로 일정하며, 향상도는 모두 1이상임을 확인할 수 있다.

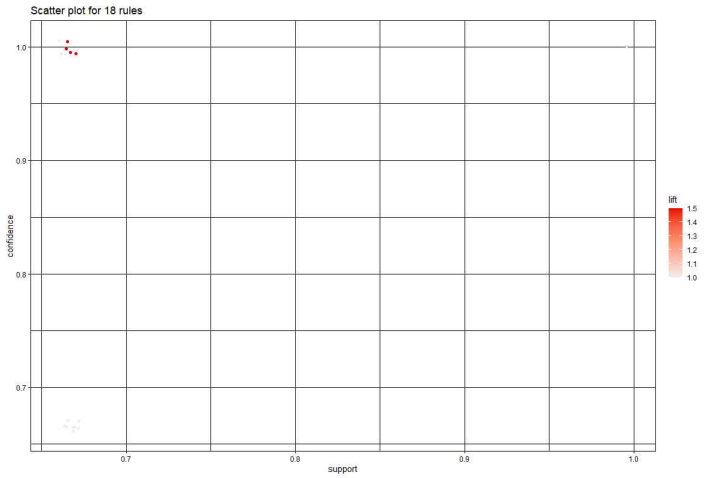
```
> inspect(ps_rule)
     lhs                                rhs                 support   confidence coverage  lift count
[1]  {}                              => {Ice Cream}         1.0000000 1          1.0000000 1.0  3
[2]  {Skim Milk}                     => {Swiss Cheese}      0.3333333 1          0.3333333 1.5  1
[3]  {Skim Milk}                     => {Cake}              0.3333333 1          0.3333333 1.5  1
[4]  {Skim Milk}                     => {Ice Cream}         0.3333333 1          0.3333333 1.0  1
[5]  {Cheddar Cheese}                => {Pancakes}          0.3333333 1          0.3333333 3.0  1
[6]  {Pancakes}                      => {Cheddar Cheese}    0.3333333 1          0.3333333 3.0  1
[7]  {Cheddar Cheese}                => {Frozen Pizza}      0.3333333 1          0.3333333 1.5  1
[8]  {Cheddar Cheese}                => {Low Fat Milk}      0.3333333 1          0.3333333 1.5  1
[9]  {Cheddar Cheese}                => {Cake}              0.3333333 1          0.3333333 1.5  1
[10] {Cheddar Cheese}                => {Ice Cream}         0.3333333 1          0.3333333 1.0  1
[11] {Pancakes}                      => {Frozen Pizza}      0.3333333 1          0.3333333 1.5  1
[12] {Pancakes}                      => {Low Fat Milk}      0.3333333 1          0.3333333 1.5  1
[13] {Pancakes}                      => {Cake}              0.3333333 1          0.3333333 1.5  1
[14] {Pancakes}                      => {Ice Cream}         0.3333333 1          0.3333333 1.0  1
[15] {Swiss Cheese}                  => {Ice Cream}         0.6666667 1          0.6666667 1.0  2
[16] {Frozen Pizza}                  => {Low Fat Milk}      0.6666667 1          0.6666667 1.5  2
[17] {Low Fat Milk}                  => {Frozen Pizza}      0.6666667 1          0.6666667 1.5  2
[18] {Frozen Pizza}                  => {Ice Cream}         0.6666667 1          0.6666667 1.0  2
[19] {Low Fat Milk}                  => {Ice Cream}         0.6666667 1          0.6666667 1.0  2
[20] {Cake}                          => {Ice Cream}         0.6666667 1          0.6666667 1.0  2
[21] {Skim Milk,Swiss Cheese}        => {Cake}              0.3333333 1          0.3333333 1.5  1
[22] {Cake,Skim Milk}                => {Swiss Cheese}      0.3333333 1          0.3333333 1.5  1
```

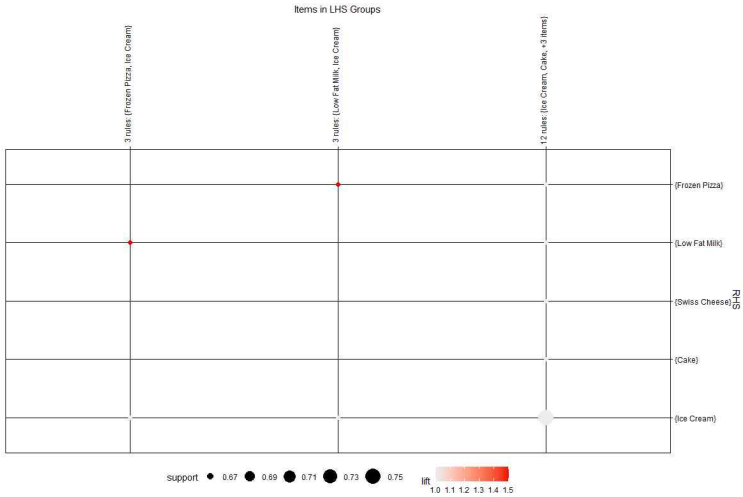
다음은 최소 지지도와 신뢰도를 설정하여 apriori 알고리즘을 돌린 결과이다.

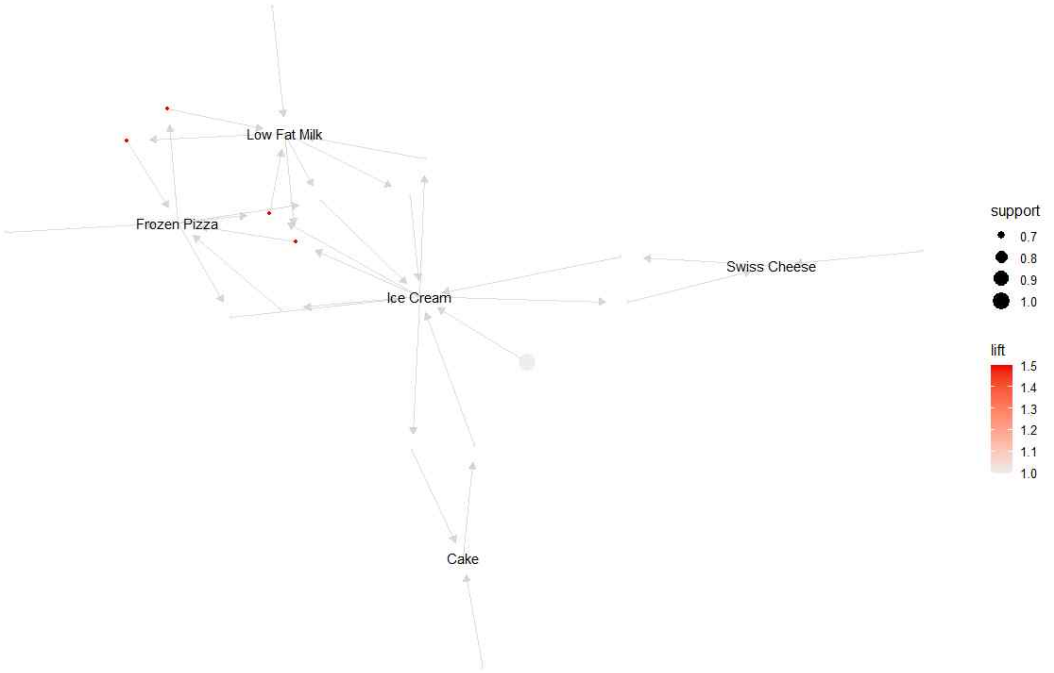
```
> ps_rule <- apriori(ps.trans, parameter = list(support = 0.4, confidence = 0.5))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.5    0.1    1 none FALSE            TRUE       5     0.4      1     10  rules TRUE

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

Absolute minimum support count: 1

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[8 item(s), 3 transaction(s)] done [0.00s].
sorting and recoding items ... [5 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [18 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].

> summary(ps_rule)
set of 18 rules

rule length distribution (lhs + rhs):sizes
 1  2  3
 5 10  3

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.250   2.000   1.889   2.000   3.000

summary of quality measures:
    support         confidence        coverage          lift           count
 Min.   :0.6667   Min.   :0.6667   Min.   :0.6667   Min.   :1.000   Min.   :2.000
 1st Qu.:0.6667   1st Qu.:0.6667   1st Qu.:0.6667   1st Qu.:1.000   1st Qu.:2.000
 Median :0.6667   Median :1.0000   Median :0.8333   Median :1.000   Median :2.000
 Mean   :0.6852   Mean   :0.8519   Mean   :0.8333   Mean   :1.111   Mean   :2.056
 3rd Qu.:0.6667   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:2.000
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.500   Max.   :3.000

mining info:
     data ntransactions support confidence
 ps.trans             3     0.4        0.5
```

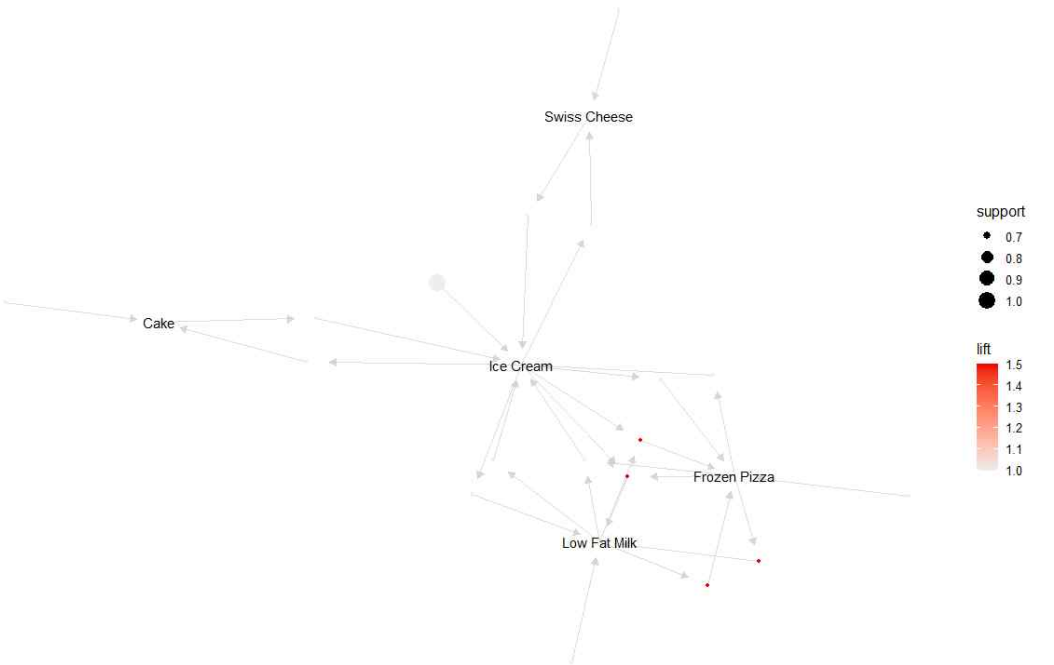
그 결과 18개의 규칙이 생성됨을 볼 수 있는데, 처음 설정 없이 돌린 것과 비교하면 많이 줄어든 것을 알 수 있다. 이 경우 최소 지지도와 신뢰도가 0.667 임을 알 수 있고, 향상도는 모두 1 이상임을 알 수 있다.

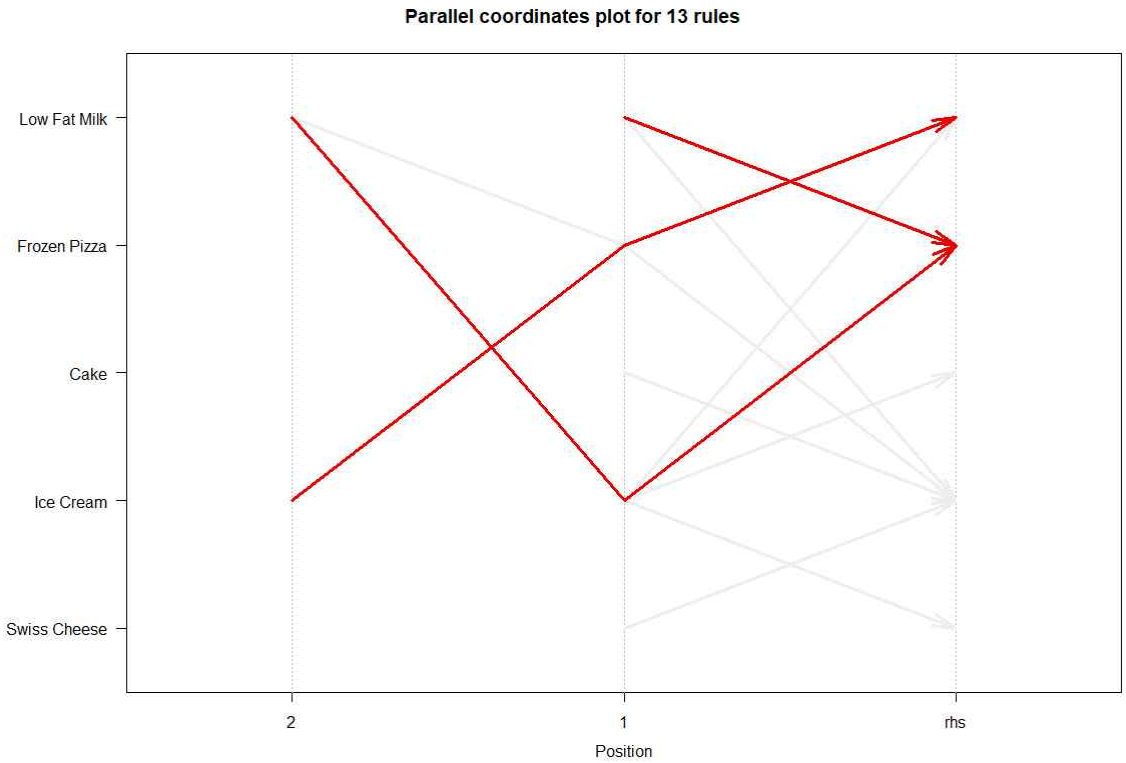
다음은 연관규칙의 시각화를 나타낸 그래프이다. 그래프는 연관규칙을 이용해 지지도와 신뢰도를 이용한 산포도를 나타낸다. 빨간색에 가까울수록 향상도가 높다.






Low Fat Milk
Frozen Pizza
Swiss Cheese
Ice Cream
Cake
support
0.7
0.8
0.9
1.0
lift
1.5
1.4
1.3
1.2
1.1
1.0


Swiss Cheese
Cake
Ice Cream
Frozen Pizza
Low Fat Milk
support
0.7
0.8
0.9
1.0
lift
1.5
1.4
1.3
1.2
1.1
1.0



다음은 각 물품 간의 연관관계를 나타낸 그래프이다. 가로축의 숫자는 물품의 수를 의미한다. 이를 주어진 prodhierarchy 데이터를 이용하여 prodsales와 연결하여 해석해볼 수 있다.

	Product	ParentProd	level
1	Whole Milk	Milk	1
2	Low Fat Milk	Milk	1
3	Skim Milk	Milk	1
4	Swiss Cheese	Cheese	1
5	Cheddar Cheese	Cheese	1
6	Waffles	Breakfast	1
7	Pancakes	Breakfast	1
8	Frozen Pizza	Dinner	1
9	Ice Cream	Dessert	1
10	Cake	Dessert	1
11	Milk	Dairy Products	2
12	Cheese	Dairy Products	2
13	Breakfast	Frozen Foods	2
14	Dinner	Frozen Foods	2
15	Dessert	Frozen Foods	2

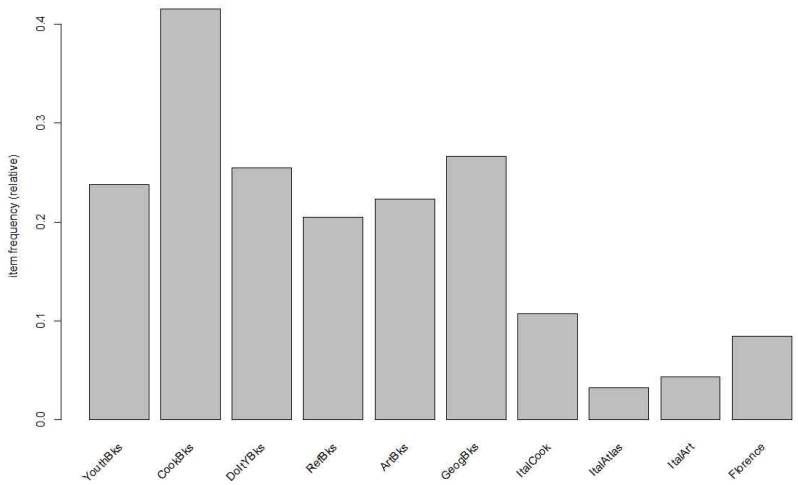
< 2. CharlesBookClub 데이터를 사용하여 분석 진행 >

```
> summary(all.books.df)
      Seq.             ID.            Gender              M                R                F             FirstPurch
 Min.   :   1   Min.   :   25   Min.   :0.0000   Min.   : 15.0   Min.   : 2.00   Min.   : 1.000   Min.   : 2.00
 1st Qu.:1001   1st Qu.: 8253   1st Qu.:0.0000   1st Qu.:129.0   1st Qu.: 8.00   1st Qu.: 1.000   1st Qu.:12.00
 Median :2000   Median :16581   Median :1.0000   Median :208.0   Median :12.00   Median : 2.000   Median :20.00
 Mean   :2000   Mean   :16595   Mean   :0.7045   Mean   :208.1   Mean   :13.39   Mean   : 3.833   Mean   :26.51
 3rd Qu.:3000   3rd Qu.:24838   3rd Qu.:1.0000   3rd Qu.:283.0   3rd Qu.:16.00   3rd Qu.: 6.000   3rd Qu.:36.00
 Max.   :4000   Max.   :32977   Max.   :1.0000   Max.   :479.0   Max.   :36.00   Max.   :12.000   Max.   :99.00
    ChildBks         YouthBks         CookBks          DoItYBks         RefBks           ArtBks
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000
 Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000   Median :0.000
 Mean   :0.6398   Mean   :0.3048   Mean   :0.7312   Mean   :0.3508   Mean   :0.2562   Mean   :0.289
 3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.000
 Max.   :7.0000   Max.   :5.0000   Max.   :7.0000   Max.   :5.0000   Max.   :4.0000   Max.   :5.000
    GeogBks          ItalCook         ItalAtlas         ItalArt          Florence       Related.Purchase
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000   Min.   :0.000
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.000
 Median :0.0000   Median :0.0000   Median :0.0000   Median :0.00000   Median :0.0000   Median :0.000
 Mean   :0.3875   Mean   :0.1252   Mean   :0.0375   Mean   :0.04575   Mean   :0.0845   Mean   :0.885
 3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:1.000
 Max.   :6.0000   Max.   :3.0000   Max.   :2.0000   Max.   :2.00000   Max.   :1.0000   Max.   :8.000
     Mcode            Rcode            Fcode        Yes_Florence     No_Florence
 Min.   :1.000   Min.   :1.00   Min.   :1.000   Min.   :0.0000   Min.   :0.0000
 1st Qu.:4.000   1st Qu.:3.00   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:1.0000
 Median :5.000   Median :3.00   Median :2.000   Median :0.0000   Median :1.0000
 Mean   :4.281   Mean   :3.17   Mean   :2.086   Mean   :0.0845   Mean   :0.9155
 3rd Qu.:5.000   3rd Qu.:4.00   3rd Qu.:3.000   3rd Qu.:0.0000   3rd Qu.:1.0000
 Max.   :5.000   Max.   :4.00   Max.   :3.000   Max.   :1.0000   Max.   :1.0000
```

다음은 주어진 데이터의 정보이다. 주어진 데이터를 이진화 데이터로 변환하기 위해 아래와 같이 진행하였다. 변환 이후 transactions도 진행하였다.

```
> count.books.df<-all.books.df[,8:18]
> incid.books.df<-ifelse(count.books.df>0,1,0)
> incid.books.mat<-as.matrix(incid.books.df[,-1])
> books.trans<-as(incid.books.mat,"transactions")
```

itemFrequencyPlot를 이용하여 나타내면 다음과 같다. 가장 많이 팔린 물품은 cookbks임을 알 수 있다.



```
> rules<-apriori(books.trans, parameter=list(support=200/4000,confidence=0.5,target="rules"))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.5    0.1    1 none FALSE            TRUE       5    0.05      1     10  rules TRUE

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

Absolute minimum support count: 200

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[10 item(s), 4000 transaction(s)] done [0.00s].
sorting and recoding items ... [8 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [21 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

다음은 aprior 알고리즘을 이용한 결과이다. 최소 지지도와 신뢰도를 정하여 진행하였다.

```
> summary(rules)
set of 21 rules

rule length distribution (lhs + rhs):sizes
 2  3
 6 15

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   2.000   3.000   2.714   3.000   3.000

summary of quality measures:
    support         confidence        coverage            lift           count
 Min.   :0.05150   Min.   :0.5067   Min.   :0.0695   Min.   :1.220   Min.   :206.0
 1st Qu.:0.05525   1st Qu.:0.5331   1st Qu.:0.0925   1st Qu.:1.637   1st Qu.:221.0
 Median :0.07450   Median :0.6758   Median :0.1045   Median :1.848   Median :298.0
 Mean   :0.08657   Mean   :0.6582   Mean   :0.1357   Mean   :1.821   Mean   :346.3
 3rd Qu.:0.08375   3rd Qu.:0.7673   3rd Qu.:0.1610   3rd Qu.:2.022   3rd Qu.:335.0
 Max.   :0.16875   Max.   :0.8400   Max.   :0.2667   Max.   :2.265   Max.   :675.0

mining info:
        data ntransactions support confidence
 books.trans          4000    0.05        0.5
```

총 21개의 규칙이 만들어진 것을 알 수 있다. 최소 지지도는 0.05임을 알 수 있고, 최소 신뢰도 또한 0.05임을 알 수 있다. 향상도는 모두 1 이상임을 알 수 있다. 이때 향상도는 높은 순서대로 아래의 사진에서 확인할 수 있다.

```
> inspect(sort(rules,by="lift"))
     lhs                      rhs         support confidence coverage lift     count
[1]  {DoItYBks,GeogBks}    => {YouthBks} 0.05450 0.5396040  0.10100  2.264864 218
[2]  {CookBks,GeogBks}     => {YouthBks} 0.08025 0.5136000  0.15625  2.155719 321
[3]  {CookBks,RefBks}      => {DoItYBks} 0.07450 0.5330948  0.13975  2.092619 298
[4]  {YouthBks,GeogBks}    => {DoItYBks} 0.05450 0.5215311  0.10450  2.047227 218
[5]  {YouthBks,CookBks}    => {DoItYBks} 0.08375 0.5201863  0.16100  2.041948 335
[6]  {YouthBks,RefBks}     => {CookBks}  0.06825 0.8400000  0.08125  2.021661 273
[7]  {YouthBks,DoItYBks}   => {GeogBks}  0.05450 0.5278450  0.10325  1.978801 218
[8]  {YouthBks,DoItYBks}   => {CookBks}  0.08375 0.8111380  0.10325  1.952197 335
[9]  {DoItYBks,RefBks}     => {CookBks}  0.07450 0.8054054  0.09250  1.938400 298
[10] {RefBks,GeogBks}      => {CookBks}  0.06450 0.7889908  0.08175  1.898895 258
[11] {YouthBks,GeogBks}    => {CookBks}  0.08025 0.7679426  0.10450  1.848237 321
[12] {DoItYBks,GeogBks}    => {CookBks}  0.07750 0.7673267  0.10100  1.846755 310
[13] {YouthBks,ArtBks}     => {CookBks}  0.05150 0.7410072  0.06950  1.783411 206
[14] {DoItYBks,ArtBks}     => {CookBks}  0.05300 0.7114094  0.07450  1.712177 212
[15] {RefBks}              => {CookBks}  0.13975 0.6825397  0.20475  1.642695 559
[16] {ArtBks,GeogBks}      => {CookBks}  0.05525 0.6800000  0.08125  1.636582 221
[17] {YouthBks}            => {CookBks}  0.16100 0.6757608  0.23825  1.626380 644
[18] {DoItYBks}            => {CookBks}  0.16875 0.6624141  0.25475  1.594258 675
[19] {ItalCook}            => {CookBks}  0.06875 0.6395349  0.10750  1.539193 275
[20] {GeogBks}             => {CookBks}  0.15625 0.5857545  0.26675  1.409758 625
[21] {ArtBks}              => {CookBks}  0.11300 0.5067265  0.22300  1.219558 452
```

한편 주어진 데이터를 이용하여 협업 필터링을 진행하였다. 다음은 사용자 기반 협업 필터링이다. 우선 주어진 데이터에서 책의 구매와 관련된 변수들만 남겨주었다. 이후 0인 부분은 NA로 , 5 이상인 것은 모두 5로 조정해주었다.

```
> library(recommenderlab)
> count <- read.csv("C:/project/CharlesBookClub2.csv", stringsAsFactors = FALSE)
> count
   ChildBks YouthBks CookBks DoItYBks RefBks ArtBks GeogBks ItalCook ItalAtlas ItalArt
1        NA        1       1       NA      NA     NA      NA       NA        NA      NA
2        NA       NA      NA       NA      NA     NA      NA       NA        NA      NA
3         2        1       2       NA       1     NA       1        1        NA      NA
4        NA       NA      NA       NA      NA     NA      NA       NA        NA      NA
5        NA       NA      NA       NA      NA     NA      NA       NA        NA      NA
6        NA       NA      NA       NA      NA     NA      NA       NA        NA      NA
7        NA       NA      NA       NA      NA     NA       1       NA        NA      NA
8         1       NA      NA       NA      NA     NA      NA       NA        NA      NA
9        NA       NA      NA       NA      NA     NA      NA       NA        NA      NA
10       NA       NA       1       NA      NA     NA      NA       NA        NA      NA
11       NA       NA       1       NA      NA     NA      NA       NA        NA      NA
12       NA       NA      NA       NA      NA      1      NA       NA        NA      NA
13       NA       NA      NA       NA      NA     NA      NA       NA        NA      NA
14        1       NA      NA       NA      NA     NA      NA       NA        NA      NA
15       NA       NA      NA       NA      NA     NA      NA       NA        NA      NA
16        2       NA       3       NA       1      1       2        1         1       1
17        2        1       2       NA      NA      1      NA        1        NA      NA
18       NA       NA       1       NA      NA     NA      NA       NA        NA      NA
19       NA       NA       2        2       1      1      NA       NA        NA      NA
20        3       NA       2       NA      NA      2       3       NA        NA      NA
21       NA       NA      NA       NA      NA     NA      NA       NA        NA      NA
22       NA       NA      NA       NA      NA     NA      NA       NA        NA      NA
```

```
> count_matrix <- as(as(count, "matrix"), 'realRatingMatrix')
> UBCF <- Recommender(count_matrix, method = 'UBCF', param = list(method = 'cosine')) # 유사도 = cosine
> tail(count)
     ChildBks YouthBks CookBks DoItYBks RefBks ArtBks GeogBks ItalCook ItalAtlas ItalArt
3995       NA       NA       NA       NA     NA      1       1       NA        NA      NA
3996       NA       NA       NA       NA     NA     NA      NA       NA        NA      NA
3997        1        1        2        2      2     NA       1       NA        NA      NA
3998       NA       NA       NA       NA     NA     NA      NA       NA        NA      NA
3999        1        1        3        1     NA     NA      NA       NA        NA      NA
4000       NA       NA       NA       NA     NA     NA       1       NA        NA      NA
> who <- 3999 # 3999번째 사람
> head(as(predict(UBCF, count_matrix[who, ], type = 'ratings'), 'list')[[1]])
   RefBks    ArtBks   GeogBks  ItalCook ItalAtlas   ItalArt
 1.350000  1.597436  1.950000  1.375000  1.500000  1.125000
> as(predict(UBCF, count_matrix[who, ], type = 'topNList', n = 3), 'list')
[[1]]
[1] "GeogBks"   "ArtBks"    "ItalAtlas"
```

이후 코사인 유사도를 이용하여 계산한 결과 위와 같은 결과가 나왔다. 만약 3999번째의 사람이 다음 책을 추천 받는다면, GeogBks, ArtBks , ItalAtlas 중 하나일 가능성이 높다는 것을 알 수 있다.

< 3. assocs 데이터를 이용하여 시차 연관성 분석 진행 >

시차연관성 규칙 발견에서 쓰이는 자료는 품목 구매목록에서 '구매순서 자료'가 추가로 포함된다는 것이 일반 연관성 규칙 발견과 크게 다른 점이다. 시차연관성 규칙에서의 주된 관심사는 '빈도가 높은 거래 품목 계열'을 찾는 것이다. 이때 구매 순서 데이터를 이용함으로써 일반 연관성 규칙에서 보다 좀 더 정교한 연관성 규칙을 얻을 수 있다.

주어진 assocs 데이터는 아래와 같이 구성되어 있다. 이 데이터는 0,1,2 등으로 된 고객이 0일, 1일, 2일 등에 구매한 구매목록으로서 구성되어 있다. 즉 첫 번째 열은 고객 ID, 두 번째 열은 Time으로 거래 순서 정보를 제공한다. 세 번째 열은 product로 구매한 품목임을 알 수 있다.

	CUSTOMER	TIME	PRODUCT
1	0	0	hering
2	0	1	corned_b
3	0	2	olives
4	0	3	ham
5	0	4	turkey
6	0	5	bourbon
7	0	6	ice_crea
8	1	0	baguette
9	1	1	soda
10	1	2	hering
11	1	3	cracker
12	1	4	heineken
13	1	5	olives
14	1	6	corned_b
15	2	0	avocado
16	2	1	cracker
17	2	2	artichok
18	2	3	heineken
19	2	4	ham
20	2	5	turkey

우선 주어진 assocs 데이터는 sas 데이터이므로, text 파일로 변환해주었다.

```
proc export data=TMP1.assocs
        outfile='C:/project/assocs.txt'
        dbms=tab replace;
run;
```

변환해준 데이터를 불러왔는데, 이때 중복구매 한 아이디 당 중복 구매는 없었으므로 size 변수를 모두 1로 추가하였다.

```
> # basket 형식의 assocsSeq.txt 불러오기
> X=read_baskets(con ="C:/project/assocsSeq.txt", info = c("sequenceID","eventID","SIZE"))
> # 자료 보기
> as(X, 'data.frame')
         items sequenceID eventID SIZE
1     {hering}          1       1    1
2   {corned_b}          1       2    1
3     {olives}          1       3    1
4        {ham}          1       4    1
5     {turkey}          1       5    1
6    {bourbon}          1       6    1
7   {ice_crea}          1       7    1
8   {baguette}          2       1    1
9       {soda}          2       2    1
10    {hering}          2       3    1
11   {cracker}          2       4    1
12  {heineken}          2       5    1
13    {olives}          2       6    1
14  {corned_b}          2       7    1
15   {avocado}          3       1    1
16   {cracker}          3       2    1
17  {artichok}          3       3    1
18  {heineken}          3       4    1
19       {ham}          3       5    1
20    {turkey}          3       6    1
21  {sardines}          3       7    1
22    {olives}          4       1    1
23   {bourbon}          4       2    1
24      {coke}          4       3    1
25    {turkey}          4       4    1
```

다음은 지지도가 0.1 이상인 시퀀스를 찾는 것이다. 결과는 다음과 같다.

시퀀스는 총 183개이고, heiken이 52개로 가장 많은 것을 알 수 있다.

```
> seq = cspade(X)
> summary(seq)
set of 183 sequences with

most frequent items:
heineken   olives   hering  cracker     soda  (Other)
      52       42       41       36       28      261

most frequent elements:
{heineken}   {olives}   {hering}  {cracker}     {soda}    (Other)
        52         42         41         36         28        261

element (sequence) size distribution:
sizes
 1  2  3  4  5
19 81 57 22  4

sequence length distribution:
lengths
 1  2  3  4  5
19 81 57 22  4

summary of quality measures:
    support
 Min.   :0.1019
 1st Qu.:0.1124
 Median :0.1199
 Mean   :0.1563
 3rd Qu.:0.1479
 Max.   :0.5994

includes transaction ID lists: FALSE

mining info:
 data ntransactions nsequences support
    X          7007       1001     0.1
> subseq=head(sort(seq, by="support"),25)
```

다음은 지지도가 큰 상위 25개 시퀀스를 찾는 과정이다.

```
> #지지도 상위 25개
> subseq=head(sort(seq, by="support"),25)
> as(subseq,  "data.frame")
                    sequence   support
11              <{heineken}> 0.5994006
9                <{cracker}> 0.4875125
12                <{hering}> 0.4855145
14                <{olives}> 0.4725275
5                <{bourbon}> 0.4025974
4               <{baguette}> 0.3916084
8               <{corned_b}> 0.3906094
3                <{avocado}> 0.3626374
84   <{cracker},{heineken}> 0.3366633
17                   <{soda}> 0.3176823
6                <{chicken}> 0.3146853
1                 <{apples}> 0.3136863
13              <{ice_crea}> 0.3126873
2               <{artichok}> 0.3046953
10                   <{ham}> 0.3046953
7                   <{coke}> 0.2957043
15               <{peppers}> 0.2957043
16              <{sardines}> 0.2957043
19                <{turkey}> 0.2827173
85    <{hering},{heineken}> 0.2347652
164    <{olives},{bourbon}> 0.2327672
144   <{hering},{corned_b}> 0.2287712
18                  <{steak}> 0.2267732
54      <{hering},{olives}> 0.2257742
82  <{baguette},{heineken}> 0.2247752
```

이는 heinken을 구매자가 1001명 중 약 59.9% 라는 의미를 말한다.

다음은 지지도가 0.1 이상인 시퀀스 중에서 신뢰도가 0.3 이상인 규칙을 찾는 과정이다.

```
> # 지지도 0.1 이상인 규칙들 중 신뢰도가 0.3 이상인 연관성 규칙 찾기
> rules = ruleInduction(seq, confidence = 0.3)
> summary(rules)
set of 148 sequencerules with

rule size distribution (lhs + rhs)
sizes
 2  3  4  5
65 57 22  4

rule length distribution (lhs + rhs)
lengths
 2  3  4  5
65 57 22  4

summary of quality measures:
    support         confidence          lift
 Min.   :0.1019   Min.   :0.3002   Min.   :0.5185
 1st Qu.:0.1099   1st Qu.:0.3765   1st Qu.:1.0839
 Median :0.1159   Median :0.5363   Median :1.4720
 Mean   :0.1321   Mean   :0.5999   Mean   :1.5942
 3rd Qu.:0.1311   3rd Qu.:0.8248   3rd Qu.:1.8446
 Max.   :0.3367   Max.   :1.0000   Max.   :3.3818

mining info:
 data ntransactions nsequences support confidence
    X          7007        1001     0.1        0.3
```

```
> # 신뢰도가 가장 큰 25개의 규칙 찾기
> subrules=head(sort(rules, by="confidence"), 25)
> # data.frame으로 결과보기
> T=as(subrules, 'data.frame')
> T
                                                      rule   support confidence     lift
54             <{sardines},{chicken},{coke}> => <{ice_crea}> 0.1158841  1.0000000 3.198083
56  <{sardines},{heineken},{chicken},{coke}> => <{ice_crea}> 0.1158841  1.0000000 3.198083
57         <{sardines},{heineken},{chicken}> => <{ice_crea}> 0.1158841  1.0000000 3.198083
79              <{baguette},{soda},{hering}> => <{heineken}> 0.1128871  1.0000000 1.668333
82               <{soda},{olives},{cracker}> => <{heineken}> 0.1048951  1.0000000 1.668333
83               <{soda},{hering},{cracker}> => <{heineken}> 0.1128871  1.0000000 1.668333
84              <{soda},{bourbon},{cracker}> => <{heineken}> 0.1048951  1.0000000 1.668333
85             <{baguette},{soda},{cracker}> => <{heineken}> 0.1128871  1.0000000 1.668333
88    <{baguette},{soda},{hering},{cracker}> => <{heineken}> 0.1128871  1.0000000 1.668333
89     <{soda},{olives},{bourbon},{cracker}> => <{heineken}> 0.1048951  1.0000000 1.668333
90              <{soda},{olives},{bourbon}> => <{heineken}> 0.1048951  1.0000000 1.668333
123              <{baguette},{soda},{hering}> => <{cracker}> 0.1128871  1.0000000 2.051230
124               <{soda},{olives},{bourbon}> => <{cracker}> 0.1048951  1.0000000 2.051230
135            <{sardines},{heineken},{chicken}> => <{coke}> 0.1158841  1.0000000 3.381757
53            <{sardines},{heineken},{coke}> => <{ice_crea}> 0.1158841  0.9914530 3.170749
55             <{heineken},{chicken},{coke}> => <{ice_crea}> 0.1158841  0.9914530 3.170749
72                        <{soda},{cracker}> => <{heineken}> 0.2177822  0.9909091 1.653167
78                    <{baguette},{hering}> => <{heineken}> 0.2087912  0.9500000 1.584917
92                   <{avocado},{artichok}> => <{heineken}> 0.1958042  0.9468599 1.579678
48                   <{sardines},{chicken}> => <{ice_crea}> 0.1158841  0.9133858 2.921084
132                     <{sardines},{chicken}> => <{coke}> 0.1158841  0.9133858 3.088849
87           <{baguette},{hering},{cracker}> => <{heineken}> 0.1128871  0.9112903 1.520336
46                  <{sardines},{heineken}> => <{ice_crea}> 0.1178821  0.9076923 2.902875
131                    <{sardines},{heineken}> => <{coke}> 0.1168831  0.9000000 3.043581
47                     <{sardines},{coke}> => <{ice_crea}> 0.1168831  0.8931298 2.856303
```

신뢰도가 0.3 이상인 상위 25개의 규칙들은 신뢰도가 대부분 0.9 이상임을 알 수 있다.

이를 몇 개만 뽑아 해석해보자면 다음과 같다.

72번째 규칙 : soda를 사고 cracker을 산 고객의 약 21.8%는 다음에 heineken을 산다.

47번째 규칙 : sardines를 사고 coke을 산 고객의 약 11.7%는 다음에 ice cream을 산다.

53번째 규칙 : ice cream 만 산 경우보다 sardines, heineken, chicken을 산 고객이 ice cream을 구매할 확률이 약 3.1배 더 높다.

디스커션

적절한 신뢰도와 지지도의 최소 한계점은 보통 어느 정도로 잡는지 궁금합니다.