연관분석(Association Analysis)

http://ocw.ulsan.ac.kr/CourseLectures.aspx?CollCd=11161&DeptCd=11178&CourseNo=20101G0298501
12주차 연관성규칙

하나의 거래나 사건에 포함된 항목 간의 관련성을 파악하여 둘 이상의 항목들로 구성된 연관성 규칙을 도출하는 탐색적인 분석 방법

"장바구니 분석"

연관성규칙은 지지도(support), 신뢰도(confidence), 향상도(lift)를 평가척도로 사용

연관분석은 구매패턴을 분석하여 고객을 대상으로 상품을 추천하거나 프로모션 및 마케팅 전략을 수립하는데 활용

연관분석 특징

- 사건의 연관규칙을 찾는 데이터마이닝 기법
- y변수가 없으며, 비지도학습에 의한 패턴 분석 방법
- 거래 사실이 기록된 트랜잭션(Transaction)형식의 데이터 셋을 이용
- 사건과 사건 간의 연관성을 찾는 방법
- 예) 기저귀와 맥주(Diapers vs. Beer) 이야기: Karen Heath는 1992년 맥주와 기저귀의 상관관계 발견
- 지지도(제품의 동시 구매패턴), 신뢰도(A제품 구매 시 B제품 구매패턴), 향상도(A제품과 B제품간의 상관성)을 연관규칙의 평가도구로 사용
- 활용분야: 상품구매 규칙을 통한 구매패턴 예측(상품 연관성)

연관분석 시행 절차
1단계: 거래 내역 데이터를 대상으로 트랜잭션 객체 생성
2단계: 품목(Item)과 트랜잭션 ID 관찰
3단계: 평가 척도(지지도, 신뢰도, 향상도)를 이용한 연관규칙(rule) 발견
4단계: 연관분석 결과에 대한 시각화
5단계: 연관분석 결과 해설 및 업무 적용

2.1 연관규칙 평가척도

(1) 지지도(support)

전체에 대한 품목A와 품목B가 동시에 일어나는 확률

Support = P(A∩B) = 품목 A와 품목 B가 동시에 포함된 거래 수 / 전체 거래수

일반적으로 지지도가 낮다는 의미는 A와 B를 동시에 구매하는 거래가 자주 발생하지 않음을 의미
Support(A→B) 와 Support(B → A)가 상호 대칭적으로 서로 같은 값을 가진다.

(2) 신뢰도(confidence)

품목 A가 구매될 때 품목 B가 동시에 구매되는 경우의 조건부확률

Confidence (A → B) = P(B|A) = 품목 A와 품목 B가 동시에 포함된 거래 수 / 품목 A를 포함한 거래수

지지도는 상호 대칭적으로 서로 같은 값을 가지기 때문에 포함 비중이 낮은 경우에는 연관성을 판단하는데 어려움이 있어 이를 보완한 것이 신뢰도
품목 A가 포함된 거래 중에서 품목 B를 포함한 거래의 비율

(3) 향상도(lift)

하위 항목들이 독립에서 얼마나 벗어나는지의 정도를 측정한 값

Lift(A → B) = 신뢰도 / 품목 B를 포함한 거래율

지지도 또는 신뢰도가 높은 연관성 규칙 중에서 우연히 연관성이 높게 보이는 것들이 나타날 수 있는데 이를 보완하기 위해서 향상도가 사용된다.

향상도가 1에 가까우면 두 상품은 서로 독립적
향상도가 1보다 작으면 두 상품은 음의 상관성
향상도가 1보다 크면 두 상품은 양의 상관성

연관규칙에 의미가 있으려면 향상도가 1보다 큰 값이어야 한다.

향상도의 값이 클수록 상품 간의 연관성이 높다고 볼 수 있다.

예시) 상품거래 트랜잭션:
T1: 라면, 맥주, 우유
T2: 라면, 고기, 우유
T3: 라면, 과일, 고기
T4: 고기, 맥주, 우유
T5: 라면, 고기, 우유
T6: 과일, 우유

[표 16.2] 연관규칙의 평가척도 결과

상품A → 상품B	지지도	신뢰도	향상도
맥주 → 고기	1/6	1/2	0.5/0.66(4/6)
라면, 맥주 → 우유	1/6	1/1	1/0.83(5/6)

지지도와 신뢰도가 높을수록 발견되는 규칙(rule)은 적어진다.

실습 (트랜잭션 객체를 대상으로 연관규칙 생성)

1단계: 연관분석을 위한 패키지 설치

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

arules 패키지

2단계: 트랜잭션(transaction) 객체 생성

```
setwd("C:/Rwork/ ")
tran <- read.transactions("tran.txt", format = "basket", sep = ",")
tran
```

read.transaction()함수: transaction 객체 생성
https://www.rdocumentation.org/packages/arules/versions/1.6-7/topics/read.transactions

3단계: 트랜잭션 데이터 보기

```
inspect(tran)
```

inspect()함수: transaction 객체 확인
https://www.rdocumentation.org/packages/arules/versions/1.6-6/topics/inspect

4단계: 규칙(rule) 발견1

```
rule <- apriori(tran, parameter = list(supp = 0.3, conf = 0.1))
inspect(rule)
```

arules패키지에서 제공하는 apriori()함수를 이용하여 트랜잭션 객체를 대상으로 규칙 발견
형식: apriori(트랜잭션 data, parameter=list(supp, conf))
Where
Default: support: 0.1, confidence: 0.8
https://www.rdocumentation.org/packages/arules/versions/1.6-7/topics/apriori

5단계: 규칙(rule) 발견2

```
rule <- apriori(tran, parameter = list(supp = 0.1, conf = 0.1))
inspect(rule)
```

2.2 트랜잭션 객체 생성

거래 데이터를 대상으로 트랜잭션 객체를 생성하기 위해서 arules패키지에서 제공되는 read.transaction()함수 사용

read.transactions(file, format=c("basket", "single"), sep=NULL, cols=NULL, rm.duplicate=FALSE, encoding="unknown")

where
file: 트랜잭션 객체를 생성할 대상의 데이터 파일명
format: 트랜잭션 데이터 셋의 형식 지정(basket 또는 single)
- basket: 여러 개의 상품(item)으로 구성된 경우 (transaction ID없이 상품으로만 구성된 경우)
- single: 트랜잭션 구분자(Transaction ID)에 의해서 상품(item)이 대응된 경우
sep: 각 상품(item)을 구분하는 구분자 지정
cols: single인 경우 읽을 컬럼 수 지정(basket은 생략)
rm.duplicates: 중복 트랜잭션 상품(item) 제거
encoding: 데이터 셋의 인코딩 방식 지정

실습 (single 트랜잭션 객체 생성)

```
setwd("C:/Rwork/")
stran <- read.transactions("demo_single", format = "single", cols = c(1, 2))
inspect(stran)
```

format="single" 속성: 한 개의 트랜잭션 구분자에 의해서 상품(item)이 연결된 경우
step 속성 제외: item은 공백으로 구분
cols 속성: 처리할 컬럼 지정

실습 (중복 트랜잭션 제거)

1단계: 트랜잭션 데이터 가져오기

```
setwd("C:/Rwork/")
stran2 <- read.transactions("single_format.csv", format = "single",
                            sep = ",", cols = c(1, 2), rm.duplicates = T)
```

중복된 트랜잭션이 존재하는 경우 해당 트랜잭션을 제거하기 위해 rm.duplicates=T 속성 지정

2단계: 트랜잭션과 상품수 확인

```
stran2
```

3단계: 요약통계량 제공

```
summary(stran2)
```

실습 (규칙 발견(생성))

1단계: 규칙 생성하기

```
astran2 <- apriori(stran2)
```

arules패키지에서 제공되는 apriori()함수는 연관규칙의 평가척도를 이용하여 규칙을 생성

2단계: 발견된 규칙 보기

```
inspect(astran2)
```

3단계: 상위 5개의 향상도를 내림차순으로 정렬하여 출력

```
inspect(head(sort(astran2, by = "lift")))
```

실습 (basket형식으로 트랜잭션 객체 생성)

```
setwd("C:/Rwork/")
btran <- read.transactions("demo_basket", format = "basket", sep = ",")
inspect(btran)
```

format="basket" 속성: 트랜잭션 구분자(transaction ID)없이 상품으로만 구성된 데이터 셋을 대상으로 트랜잭션 객체를 생성할 경우

2.3 연관규칙 시각화

arules패키지에서 제공되는 내장 데이터 Adult를 이용하여 연관규칙을 생성하고 유사한 연관규칙끼리 네트워크 형태로 시각화
https://www.rdocumentation.org/packages/arules/versions/1.6-8

실습 (Adult 데이터 셋 가져오기)

```
data(Adult)
Adult
```

더 알아보기 (Adult 데이터 셋에 관한 설명)
https://www.rdocumentation.org/packages/arules/versions/1.6-8/topics/Adult

실습 (AdultUCI 데이터 셋 보기)

```
data("AdultUCI")
str(AdultUCI)
```

실습 (Adult 데이터 셋의 요약통계량 보기)

1단계: data.frame형식으로 보기

```
adult <- as(Adult, "data.frame")
str(adult)
head(adult)
```

2단계: 요약통계량

```
summary(Adult)
```

실습 (지지도 10%와 신뢰도 80%가 적용된 연관규칙 발견)

```
ar <- apriori(Adult, parameter = list(supp = 0.1, conf = 0.8))
```

실습 (다양한 신뢰도와 지지도를 적용한 예)

1단계: 지지도를 20%로 높인 경우 1,306개 규칙 발견

```
ar1 <- apriori(Adult, parameter = list(supp = 0.2))
```

2단계: 지지도를 20%, 신뢰도 95%로 높인 경우 348개 규칙 발견

```
ar2 <- apriori(Adult, parameter = list(supp = 0.2, conf = 0.95))
```

3단계: 지지도를 30%, 신뢰도 95%로 높인 경우 124개 규칙 발견

```
ar3 <- apriori(Adult, parameter = list(supp = 0.3, conf = 0.95))
```

4단계: 지지도를 35%, 신뢰도 95%로 높인 경우 67개 규칙 발견

```
ar4 <- apriori(Adult, parameter = list(supp = 0.35, conf = 0.95))
```

5단계: 지지도를 40%, 신뢰도 95%로 높인 경우 36개 규칙 발견

```
ar5 <- apriori(Adult, parameter = list(supp = 0.4, conf = 0.95))
```

실습 (규칙 결과 보기)

1단계: 상위 6개 규칙 보기

```
inspect(head(ar5))
```

2단계: confidence(신뢰도)기준 내림차순 정렬 상위 6개 출력

```
inspect(head(sort(ar5, decreasing = T, by = "confidence")))
```

3단계: lift(향상도)기준 내림차순 정렬 상위 6개 출력

```
inspect(head(sort(ar5, by = "lift")))
```

실습 (연관규칙 시각화)

1단계 패키지 설치

```
install.packages("arulesViz")
library(arulesViz)
```

arulesViz패키지

2단계: 연관규칙 시각화

```
plot(ar3, method = "graph", control = list(type = "items"))
```

지지도, 신뢰도 조정 필요

실습 (Groceries 데이터 셋으로 연관분석)

arules패키지에서 제공되는 Groceries 데이터 셋 사용

1단계: Groceries 데이터 셋 가져오기

```
data("Groceries")
str(Groceries)
Groceries
```

더 알아보기 (Groceries 데이터 셋)
https://www.rdocumentation.org/packages/arules/versions/1.6-8/topics/Groceries

2단계: 데이터프레임으로 형 변환

```
Groceries.df <- as(Groceries, "data.frame")
head(Groceries.df)
```

3단계: 지지도 0.001, 신뢰도 0.8 적용 규칙 발견

```
rules <- apriori(Groceries, parameter = list(supp = 0.001, conf = 0.8))
```

4단계: 규칙을 구성하는 왼쪽(LHS) → 오른쪽(RHS)의 item 빈도수 보기

규칙의 표현 A(LHS) → B(RHS)

```
plot(rules, method = "grouped")
```

실습 (최대 길이가 3 이하인 규칙 생성)

```
rules <- apriori(Groceries,
                 parameter = list(supp = 0.001, conf = 0.80, maxlen = 3))
```

규칙을 구성하는 LHS와 RHS 길이를 합쳐서 3이하의 길이를 갖는 규칙 생성

실습 (Confidence(신뢰도)기준 내림차순으로 규칙 정렬)

```
rules <- sort(rules, decreasing = T, by = "confidence")
inspect(rules)
```

실습 (발견된 규칙 시각화)

```
library(arulesViz)
plot(rules, method = "graph")
```

실습 (특정 상품(Item)으로 서브 셋 작성과 시각화)

1단계: 오른쪽 item이 전지분유(whole milk)인 규칙만 서브 셋으로 작성

```
wmilk <- subset(rules, rhs %in% 'whole milk')
wmilk

inspect(wmilk)
plot(wmilk, method = "graph")
```

2단계: 오른쪽 item이 other vegetables인 규칙만 서브 셋으로 작성

```
oveg <- subset(rules, rhs %in% 'other vegetables')
oveg
```

```
inspect(oveg)
plot(oveg, method = "graph")
```

3단계: 오른쪽 item이 vegetables 단어가 포함된 규칙만 서브 셋으로 작성

```
oveg <- subset(rules, rhs %pin% 'vegetables')

oveg
inspect(oveg)
plot(oveg, method = "graph")
```

arules 패키지 내 match
%pin% uses partial matching on the table;
%ain% itemsets have to match/include all items in the table;
%oin% itemsets can only match/include the items in the table.

```
x %in% table
x %pin% table
x %ain% table
x %oin% table
```

4단계: 왼쪽 item이 butter 또는 yogurt인 규칙만 서브 셋으로 작성

```
butter_yogurt <- subset(rules, lhs %in% c('butter', 'yogurt'))
butter_yogurt
inspect(butter_yogurt)
plot(butter_yogurt, method = "graph")
```

연관 네트워크 그래프에서 타원의 크기는 지지도(조합), 색상은 향상도(관련성), 화살표는 상품(item)간의 관계를 나타낸다.

Ch16 연습문제 3 & 4번