다변량분석 보고서: 7차

Association Rule Mining

고려대학교

고려대학교 공과대학
산업경영공학부
2015170867
이종현

목차

[서론] Associative Rule Mining: 연관규칙분석

연관규칙분석이란 비지도학습의 방법론 중 하나이다. 지금껏 진행해온 지도학습은 데이터에 정답, 즉 종속변수가 포함되어 있었으므로 기존 데이터셋을 활용하여 알고리즘을 구축하여 알고리즘의 성능평가를 진행해 왔다. 하지만 연관규칙분석은 종속변수가 없기 때문에 '정답' 이 없다고 해석 할 수도 있다. 그렇기 때문에 연관규칙분석은 모델을 구축하는데 있어서 설정할 수 있는 hyperparameter에 대해서 변수들간의 연관성이 결정된다는 특징을 가진다.

연관규칙분석은 각 변수들간 얼만큼의 연관성이 있는지를 판별 할 수 있는 방법론이다. 사용한 데이터셋은 MOOC (Massive Open Online Course)를 수강 한 적이 있는 학생들에 대한 데이터이다. 어디학교를 다니는지, 무슨 수업을 수강하였는지, 언제 수강하였는지, 학생의 고유 ID는 무엇인지, 단순히 수강만 하였는지, 혹은 수강을 마치고 증명서를 받았는지, 학생의 지역이 어디인지, 학위 수준은 무엇인지, 성별은 무엇인지 등 많은 변수를 포함하고 있다. 연관규칙분석을 통해 각 변수들이 서로 얼만큼의 연관성을 가지고 있는지 알아낼 수 있지만, 가장 중요하다고 생각하는 네가지의 변수들의 연관성을 알아보고자 한다.

[Step 1] Data Preprocessing

```r
#Step1: Data Preprocessing
mooc_dataset <- read.csv("big_student_clear_third_version.csv")
Institute <- mooc_dataset[,c(2)]
Course <- mooc_dataset[,c(3)]
Region <- gsub(" ","",mooc_dataset[,c(10)])
Degree <- gsub(" ","",mooc_dataset[,c(11)])

Transaction_ID <- mooc_dataset[,c(6)]

RawTransactions <- paste(Institute, Course, Region, Degree, sep = '_')
MOOC_transactions <- paste(Transaction_ID, RawTransactions, sep = ' ')
write.csv(MOOC_transactions, file = "MOOC_User_Course.csv",
          row.names = FALSE, quote = FALSE)
```

연관규칙분석을 진행하려면 raw data를 processing 하여 transaction 데이터를 확보해야한다. 그러기 위해 먼저 기존 제공된 데이터를 불러 온 뒤, 다섯가지의 변수를 추출하고 변수명에 할당하였다.

추출된 변수명	할당된 변수명
institute (강좌 제공 기관)	Institute
course_id (강좌 코드)	Course
final_cc_cname_DI (접속 국가)	Region
LoE_DI (학위 과정)	Degree
userid_DI (사용자 아이디)	Transaction_ID

이후 paste 함수를 통해 첫 네개의 변수들의 string 값을 '_' seperator 로 사용하여 합하고, RawTransactions에 할당하였다. 5번째 함수인 Transaction_ID 를 RawTransactions과 paste 함수를 통해 합하고, 공백을 seperator로 사용한 뒤 MOOC_transactions 변수명에 할당하였다. 이 MOOC_transactions 라는 변수는 연관규칙분석에서 사용될 transaction list이다. 이 변수를 write.csv 함수를 사용하여 저장 후 실행시켜보면 아래의 파일이 생성된 것을 확인 할 수 있었다.

MOOC_User_Course

x	
MHxPC130313697 HarvardX_PH207x_India_Bachelor's	
MHxPC130237753 HarvardX_PH207x_UnitedStates_Secondary	
MHxPC130202970 HarvardX_CS50x_UnitedStates_Bachelor's	
MHxPC130223941 HarvardX_CS50x_OtherMiddleEast/CentralAsia_Secondary	
MHxPC130317399 HarvardX_PH207x_Australia_Master's	
MHxPC130191782 HarvardX_CS50x_Pakistan_Bachelor's	

첫번째 객체를 분석해보면 다음과 같다.

MHxPC130313697 = Transaction_ID
HarvardX = Institute
PH207x = Course
India = Region
Bachelor's = Degree

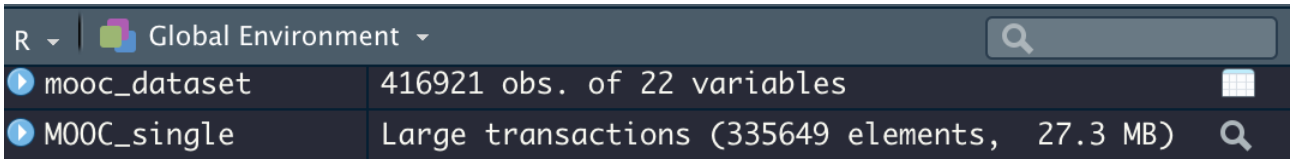
코드로 명령하였던 것 처럼 Transaction_ID 와 나머지 변수들은 공백으로 분리되어 있는 것을 볼 수 있고, 나머지 변수들은 '_'로 연결되어 있는 것을 확인 할 수 있다.

[Step 2] Data Reading

[Q1] Creating Transaction List, summarizing

```
#Step2: Data Reading
#Question 2-1
MOOC_single <- read.transactions("MOOC_User_Course.csv", format = "single",
                                 header = TRUE, cols = c(1,2),
                                 rm.duplicates = TRUE, skip = 1)
summary(MOOC_single)
str(MOOC_single)
```

앞서 생성한 transaction list 인 “MOOC_User_Course” 에 대한 연관규칙분석을 수행하기 위해 read.transaction 함수를 사용했다. 데이터의셋의 열은 하나이기에 format 은 single 로 설정하였다. 앞서 생성한 데이터셋을 보면 첫번째 열에 x라는 객체가 생긴 것을 볼 수 있는데, 이는 write.csv 함수 특성상 column name을 생성하기 때문이다. 그렇기 때문에 첫번째 열을 skip = 1 로 제외시켰다.



기존 데이터셋과 새로 생성된 transaction을 비교해보면 객체 수가 81272 줄어든 것을 확인 할 수 있다. 이는 single format transaction을 설정하게 되면 같은 Transaction_ID를 가지고 있는 데이터들은 통합이 되어 할당되기 때문에 객체 수가 줄어든 것이다. 정성적인 해석을 하자면 같은 학생이 다른 수업을 들은 것이기 때문에 학생의 한 아이디에다 수업을 전부 append 한 것이다.

```
> summary(MOOC_single)
transactions as itemMatrix in sparse format with
 335649 rows (elements/itemsets/transactions) and
 1405 columns (items) and a density of 0.0008771195

most frequent items:
    MITx_6.00x_UnitedStates_Bachelor's      MITx_6.00x_UnitedStates_Secondary
                                 14192                                   8841
          MITx_6.00x_India_Bachelor's           MITx_6.002x_India_Bachelor's
                                  7813                                   7633
HarvardX_CS50x_UnitedStates_Bachelor's                                (Other)
                                  7410                                 367749

element (itemset/transaction) length distribution:
sizes
     1      2      3      4      5      6      7      8      9     10     11     12     13
278439  43061   9997   2812    799    293    109     44     37     22     21      9      6

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   1.000   1.232   1.000  13.000

includes extended item information - examples:
                             labels
1 HarvardX_CB22x_Australia_Bachelor's
2   HarvardX_CB22x_Australia_Master's
3  HarvardX_CB22x_Australia_Secondary

includes extended transaction information - examples:
   transactionID
1 MHxPC130000002
2 MHxPC130000004
3 MHxPC130000006
```

Summary 함수를 통해 transaction list를 출력한 값은 상단의 출력창에서 확인 할 수 있다. 상단을 주목해보면, sparse format 으로 나타난 것을 알 수 있으며, 335649개의 객체와 1405개의 아이템으로 구성된 것을 알 수 있다. Density 가 약 0.00087이라는 것은 총 데이터셋의 cell 중 약 0.087%가 1의 값을 가지고 있다는 뜻이다.

출력창의 중앙을 주목해보면, most frequent items: 이 출력되어 있는 것을 볼 수 있다. 이는 cell 에 가장 많이 나타난 데이터로써, 첫번째 값을 정성적으로 풀어보면 “MITx에서 제공하는 6.00x의 학수번호를 가진 수업을 미국의 학사들이 제일 많이 수강했다” 로 해석할 수 있다.

Element (itemset/transaction) length distribution 은 한개의 객체 중 몇개의 변수가 들어있는지를 지표적으로 나타내준다. 한 학생이 한 과목만 수강한 경우가 278439 건이였으며 , 13개를 수강한 경우가 6건이었다.

[Q2-2] Vizualization of data through Wordcloud

```
#Question 2-2 Wordcloud
itemName <- itemLabels(MOOC_single)
itemCount <- itemFrequency(MOOC_single)*nrow(MOOC_single)
wordcloud(words = itemName, freq = itemCount, min.freq = 700,
          scale = c(1.2,0.2), col = brewer.pal(8,"Dark2"), random.order = FALSE)
```

데이터를 조금 더 이해하기 쉽도록 워드클라우드를 통한 시각화를 진행해보았다. Wordcloud 함수를 통해 워드클라우드를 출력 할 수 있는데, 내부에 사용되는 argument 값에 대한 설명은 다음과 같다.

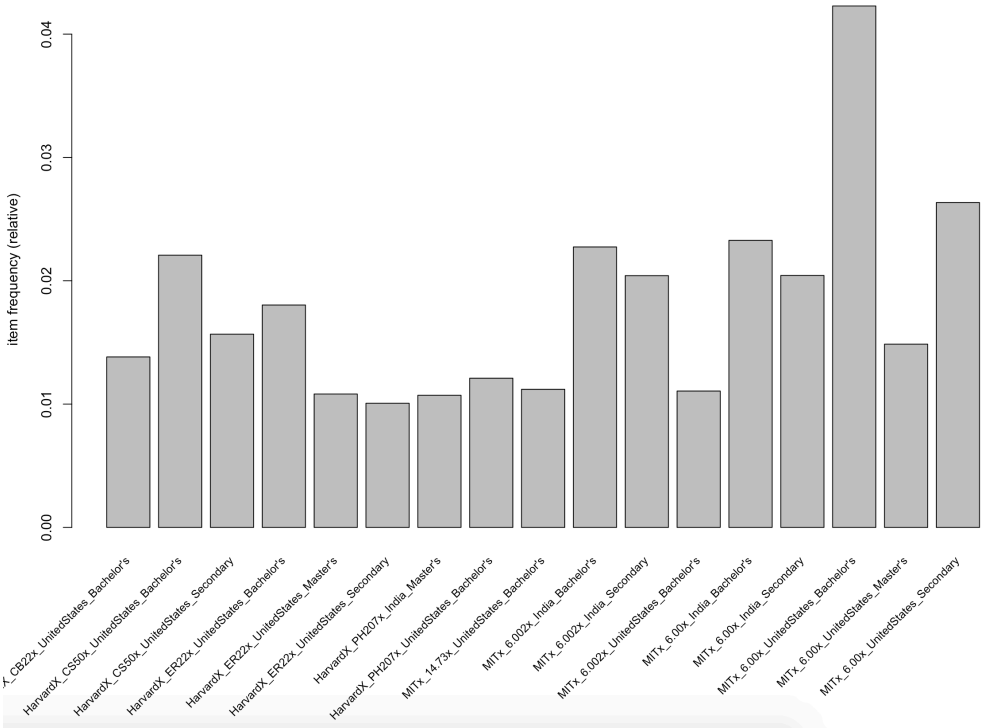
Words = 실제로 시각화할 단어들을 설정한다. Transaction list의 label을 추출하였다.
Freq = 빈도수이다. itemFrequecy 함수를 통해 transaction list의 빈도수의 백분율을 할당하였다.
Min.freq = 몇번 이상 등장해야 워드클라우드에 포함시키는지에 대한 cutoff 이다. 객체 수와 변수들의 수가 굉장히 많으므로 700으로 설정하였다.
Scale = 많이 등장하는 객체들과 적게 등장하는 객체들의 상대적 크기를 설정해준다. 1.2와 0.2로 설정하는 것이 시각적으로 만족스러운 워드클라우드를 도식한다.
Col = 워드클라우드의 색상을 지정해준다. 실습때 사용한 기본값을 사용하였다.
Random.order = 오더링을 무작위로 하는지를 boolean 값으로 지정한다. False 로 지정했다.

출력된 워드클라우드는 상단에서 확인 할 수 있다. Summary 함수로 확인하였을때 제일 많이 언급된, 즉 support 값이 제일 높은 'MITx_6.00x_UnitedStates_Bachelor's' 가 검정색으로 가장 크게 표시되었으며, 언급 횟수가 작을 수록 크기가 작은 것을 볼 수 있다. 워드클라우드의 가장 큰 장점은 정성적인 해석과 이해가 가능하다는 것이다. 머신러닝에 대한 사전지식이 없는 사람도 워드클라우드를 통해 어떤 객체가 가장 많이 있고, 비중이 높은지에 대한 해석을 할 수 있다.

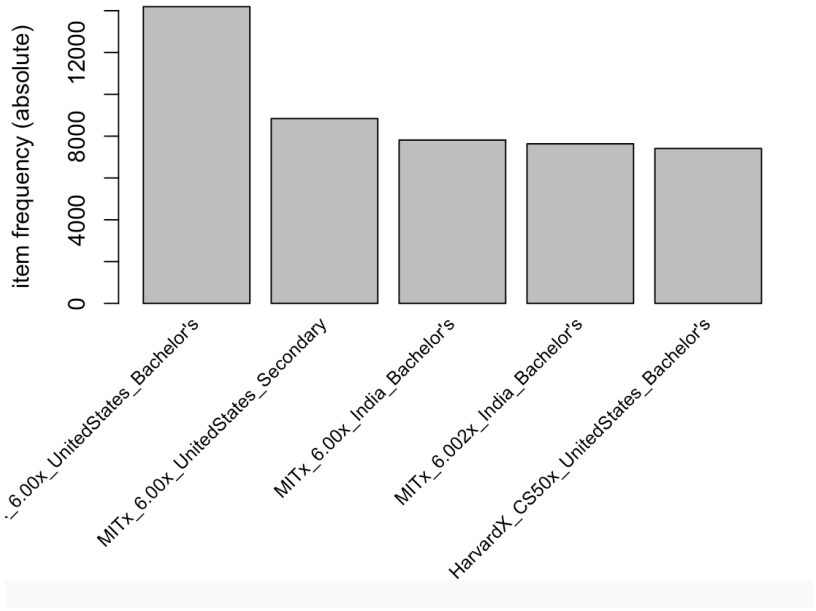
[Q2-3] Bar Chart

```
#Question 2-3 Bar chart
itemFrequencyPlot(MOOC_single, support = 0.01, cex.names = 0.8)
itemFrequencyPlot(MOOC_single, topN = 5, type = "absolute", cex.names = 0.8)
```

Transaction list에 대한 시각화는 워드클라우드 이외에도 bar plot을 사용할 수 있다. 최소 빈도 1% 이상 등장하는 Item들에 대한 barplot을 도식하였다. itemFrequencyPlot을 사용하였으며, support를 0.01로 설정하여 적어도 1% 이상 등장하는 item들만 사용하였다.



1% 이상 등장하는 Item 들은 총 17개로 나왔다. 조금 더 자세히 보기 위해 상위 5개의 아이템을 topN = 5 을 통해서 설정하여 barplot 을 도식했다.



상위 5개의 아이템은 내림차순으로:

1. MITx_6.00x_UnitedStates_Bachelor's
2. MITx_6.00x_UnitedStates_Secondary
3. MITx_6.00x_India_Bachelor's
4. MITx_6.002x_India_Bachelor's
5. HarvardX_CS50x_UnitedStates_Bachelor's

로 나왔다. 정성적인 평가를 진행해보자면, 학생들이 가장 많이 들은 강의는 MITx에서 제작한 강의들이였으며, 미국과 인도의 학부생들과 석사과정 학생들이 가장 많이 수강한 것을 알 수 있다.

[Step 3] Association Rule Construction

[Q3-1] Hyperparameter grid search

연관규칙분석에선 어떤 item set, 즉 규칙이 유용한지에 따른 3가지 평가지표를 사용 할 수 있다. 그 중 support 과 confidence 값을 hyperparameter 로 설정하여 이전에 생성한 transaction list 에 대해 생성되는 규칙의 개수를 파악하였다.

```r
#Step3: Generate Rules and Interpret Results
#Question 3-1
support_rule <- c(0.0005, 0.0015, 0.002, 0.0025)
confidence_rule <- c(0.005, 0.01, 0.1)

matrix_rules <- matrix(0,4,3)
rownames(matrix_rules) <- paste0("Support = ",support_rule)
colnames(matrix_rules) <- paste0("Confidence = ",confidence_rule)
matrix_rules

start.time <- proc.time()
for(i in 1:4){
  for(j in 1:3){
    tmp_a <- support_rule[i]
    tmp_b <- confidence_rule[j]
    cat("Support:",support_rule[i],",Confidence:",confidence_rule[j],"\n")

    rules_tmp <- apriori(MOOC_single, parameter = list(support = tmp_a, confidence = tmp_b))
    rules_tmp <- data.frame(length(rules_tmp), tmp_a, tmp_b)

    tmp_cnt <- rules_tmp[,1]
    matrix_rules[i,j] <- tmp_cnt
  }
}
end.time <- proc.time()
time <- end.time - start.time
time
```

Support 의 값은 0.0005, 0.0015, 0.002, 0.0025로 hyperparameter을 설정하였고, Confidence 의 값은 0.005, 0.01, 0.1로 설정하여 내부의 apriori 함수를 이용한 grid search 를 진행하였다. Apriori 는 설정된 hyperparameter 기준으로 실질적인 연관규칙 분석을 해주는 함수다.

```
> matrix_rules
                 Confidence = 0.005 Confidence = 0.01 Confidence = 0.1
Support = 5e-04                 240                214               103
Support = 0.0015                 73                 47                22
Support = 0.002                  63                 37                16
Support = 0.0025                 57                 31                12
```

Grid search 을 통해 hyperparameter 의 조합에 따른 생성된 규칙의 수는 상단의 출력창에서 확인 할 수 있다. 결론적으로 생성된 규칙의 수는 전부 10개 이상인 것을 확인 할 수 있었다. Hyperparameter 값이 낮을수록 더 많은 규칙을 생성하는 것을 볼 수 있다. 각 조건절 (if) 과 결과절 (then) 이 나타나야 하는 빈도수가 줄어들기 때문에 많은 규칙이 생성되는 원리이다. 하지만 많은 규칙이 생성되는 것은 무조건적으로 좋은것은 아니기 때문에, 표현하고자 하는 데이터에 따른 분석을 진행하여 적절한 hyperparameter 값을 설정해야 한다.

[Q3-2] ARM using given Hyperparamters

Support = 0.001, confidence = 0.05 로 지정하여 apriori 함수를 통한 연관규칙분석을 진행하였다.

```
#Question 3-2: fixed hyperparameters
fixed_h <- apriori(MOOC_single, parameter = list(support = 0.001, confidence = 0.05))
inspect(fixed_h)
inspect(sort(fixed_h, by = "support"))
inspect(sort(fixed_h, by = "confidence"))
inspect(sort(fixed_h, by = "lift"))
fixed_h
str(fixed_h)
```

총 51개의 규칙이 생성되었으며, 각 hyperparameter (support, confidence, lift)를 기준으로 가장 높은 hyperparameter 을 갖는 규칙을 찾기위해 sort 함수를 사용하여 정렬하였다.

```
> inspect(sort(fixed_h, by = "support"))
    lhs                                                  rhs                                                  support confidence    coverage      lift count
[1] {HarvardX_CS50x_UnitedStates_Bachelor's}  => {MITx_6.00x_UnitedStates_Bachelor's}      0.003643687 0.16504723 0.022076634  3.903462  1223
[2] {MITx_6.00x_UnitedStates_Bachelor's}      => {HarvardX_CS50x_UnitedStates_Bachelor's}  0.003643687 0.08617531 0.042282265  3.903462  1223
[3] {MITx_6.00x_India_Secondary}              => {MITx_6.002x_India_Secondary}             0.003625811 0.17745698 0.020432058  8.692828  1217
[4] {MITx_6.002x_India_Secondary}             => {MITx_6.00x_India_Secondary}              0.003625811 0.17761238 0.020414183  8.692828  1217
[5] {MITx_6.002x_India_Bachelor's}            => {MITx_6.00x_India_Bachelor's}             0.003092516 0.13598847 0.022741018  5.842109  1038
[6] {MITx_6.00x_India_Bachelor's}             => {MITx_6.002x_India_Bachelor's}            0.003092516 0.13285550 0.023277293  5.842109  1038
```

우선 support 을 기준으로 내림차순 하였을 때, 가장 큰 support 을 나타낸 규칙은
조건절: HarvardX_CS50x_UnitedStates_Bachelor's
결과절: MITx_6.00x_UnitedStates_Bachelor's
이며, support 값은 0.003644 이다.

Support 값이 0.003644 라는 것은 약 1000번마다 3번 정도 등장한다는 것이다. 또한, confidence 값 0.165이 의미하는 것은 조건절이 만족되면 조건절과 결과절이 동시에 만족될 확률이 약 16.5% 이라는 것이다. Lift 값이 3.9034 라는 것은 두 아이템이 통계적으로 독립이라고 가정하에 실제로 약 3.9배 더 해당 규칙이 발생했다는 것이다.

```
> inspect(sort(fixed_h, by = "confidence"))
    lhs                                                  rhs                                                  support     confidence coverage    lift      count
[1] {MITx_8.02x_India_Secondary}              => {MITx_6.002x_India_Secondary}             0.002800545 0.38810900 0.007215871 19.011734  940
[2] {MITx_8.02x_India_Bachelor's}             => {MITx_6.002x_India_Bachelor's}            0.002496656 0.38564197 0.006474025 16.957990  838
[3] {HarvardX_CS50x_India_Secondary}          => {MITx_6.00x_India_Secondary}              0.002681373 0.29392554 0.009122625 14.385508  900
[4] {MITx_6.002x_UnitedStates_Secondary}      => {MITx_6.00x_UnitedStates_Secondary}       0.001939526 0.28194023 0.006879210 10.703875  651
[5] {HarvardX_CS50x_India_Bachelor's}         => {MITx_6.00x_India_Bachelor's}             0.002016988 0.26918489 0.007492947 11.564270  677
[6] {MITx_6.002x_UnitedStates_Bachelor's}     => {MITx_6.00x_UnitedStates_Bachelor's}      0.002818420 0.25484914 0.011059172  6.027329  946
```

Confidence 을 기준으로 내림차순 하였을 때, 가장 큰 confidence 를 나타낸 규칙은
조건절: MITx_8.02x_India_Secondary
결과절: MITx_6.002x_India_Secondary
이며, confidence 값은 0.38811 이다.

즉, 조건절의 아이템이 발생하였을 때, 조건절의 아이템과 결과절의 아이템이 동시에 발생할 가능성이 약 38.8%가 된다는 것이다. Support 값은 0.0028이며, lift 값은 19.011이다.

```
> inspect(sort(fixed_h, by = "lift"))
     lhs                                      rhs                                      support     confidence coverage    lift      count
[1]  {MITx_8.02x_UnitedStates_Bachelor's}     => {MITx_6.002x_UnitedStates_Bachelor's} 0.001391334 0.21620370 0.006435294 19.549719  467
[2]  {MITx_6.002x_UnitedStates_Bachelor's}    => {MITx_8.02x_UnitedStates_Bachelor's}  0.001391334 0.12580819 0.011059172 19.549719  467
[3]  {HarvardX_CB22x_UnitedStates_Secondary} => {HarvardX_ER22x_UnitedStates_Secondary} 0.001540300 0.19240789 0.008005387 19.106957  517
[4]  {HarvardX_ER22x_UnitedStates_Secondary} => {HarvardX_CB22x_UnitedStates_Secondary} 0.001540300 0.15295858 0.010070043 19.106957  517
[5]  {MITx_6.002x_India_Secondary}            => {MITx_8.02x_India_Secondary}          0.002800545 0.13718622 0.020414183 19.011734  940
[6]  {MITx_8.02x_India_Secondary}             => {MITx_6.002x_India_Secondary}         0.002800545 0.38810900 0.007215871 19.011734  940
```

마지막으로 lift 을 기준으로 내림차순 하였을 때, 가장 큰 Lift 값을 나타낸 규칙은
조건절: MITx_8.02x_UnitedStates_Bachelor's
결과절: MITx_6.002x_UnitedStates_Bachelor's
이며, lift 값은 19.5497 이다.

이는 조건절과 결과절이 통계적으로 독립이라고 가정하에 실제로 약 19.5497배 더 해당 규칙이 발생했다는 것이다. Support 값은 0.00139 이며, confidence 값은 0.2162 이다.

하나의 규칙에 대한 효용성 지표를 support x confidence x lift 로 정의하여 효용성이 가장 높은 규칙을 탐색해보았다.

```
#support x confidence x lift
fixed_h_df <- DATAFRAME(fixed_h)
fixed_h_df$Perf_New <- fixed_h_df$support * fixed_h_df$confidence * fixed_h_df$lift
fixed_h_df <- fixed_h_df[order(fixed_h_df[,8],decreasing = T),]
fixed_h_df
```

앞서 정의했던 규칙들을 DATAFRAME 함수를 통해 데이터프레임으로 변환하였다. 이후, Perf_New 라는 변수를 생성하여 해당 공식의 값 (support x confidence x lift) 로 정의하여 계산한 뒤, 가장 큰 지표를 갖고 있는 규칙을 찾기 위해 내림차순으로 정렬하였다.

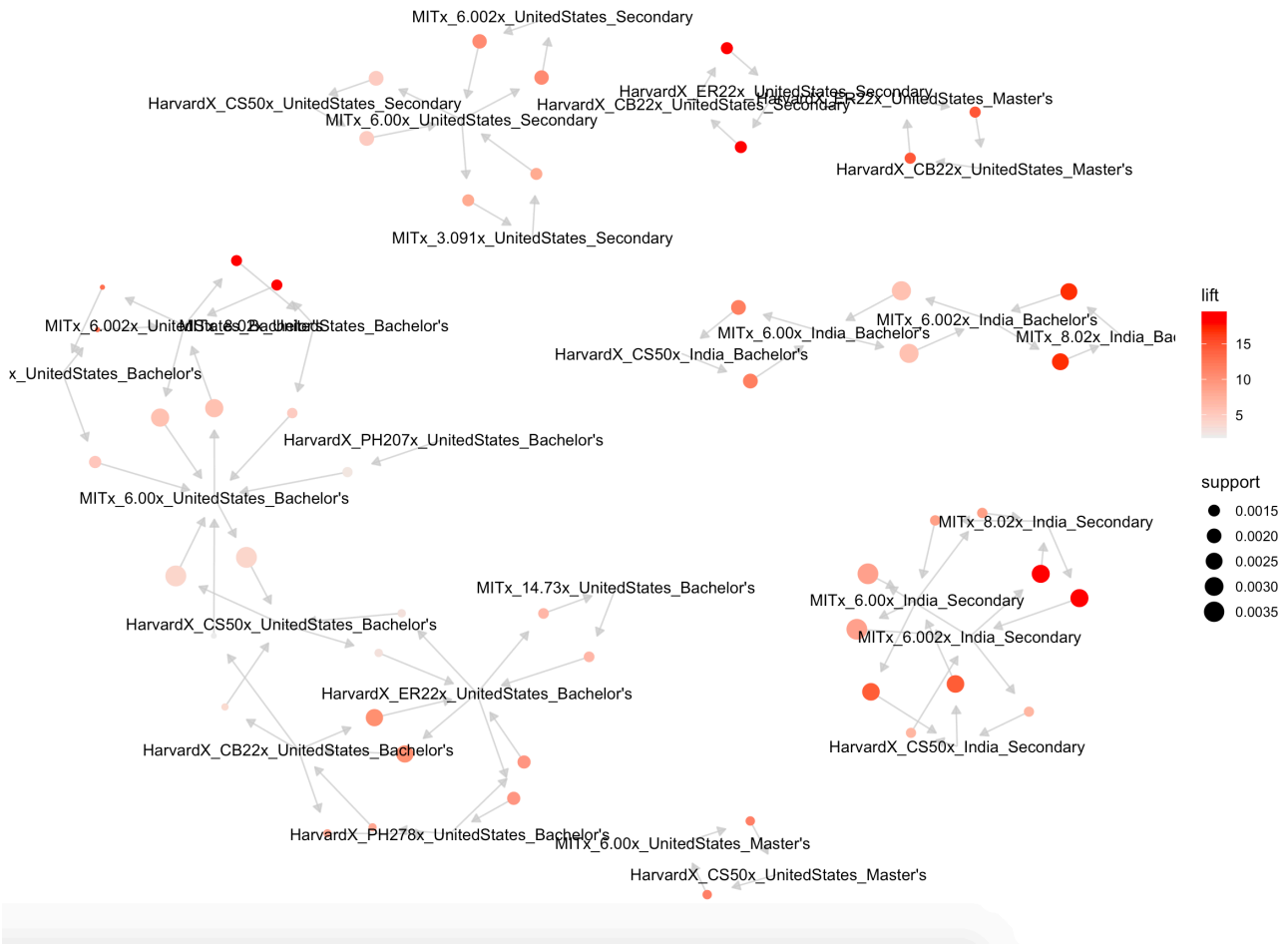
```
> fixed_h_df
                                LHS                                      RHS     support confidence    coverage      lift count     Perf_New
23             {MITx_8.02x_India_Secondary}           {MITx_6.002x_India_Secondary} 0.002800545 0.38810900 0.007215871 19.011734   940 0.0206641682
5              {MITx_8.02x_India_Bachelor's}          {MITx_6.002x_India_Bachelor's} 0.002496656 0.38564197 0.006474025 16.957990   838 0.0163274116
25          {HarvardX_CS50x_India_Secondary}            {MITx_6.00x_India_Secondary} 0.002681373 0.29392554 0.009122625 14.385508   900 0.0113375620
24             {MITx_6.002x_India_Secondary}           {MITx_8.02x_India_Secondary} 0.002800545 0.13718622 0.020414183 19.011734   940 0.0073042346
3           {HarvardX_CS50x_India_Bachelor's}          {MITx_6.00x_India_Bachelor's} 0.002016988 0.26918489 0.007492947 11.564270   677 0.0062787357
```

위의 표를 통해 가장 높은 복합평가지표를 가진 규칙은
조건절: MITx_8.02x_India_Secondary
결과절: MITx_6.002x_India_Secondary
이며,
복합평가지표값은 0.0206614 이다.

두번째로 높은 복합평가지표를 가진 규칙은
조건절: MITx_8.02x_India_Bachelor's
결과절: MITx_6.002x_India_Bachelor's
이며,
복합평가지표는 0.016327 이다.

세번째로 높은 복합평가지표를 가진 규칙은
조건절: HarvardX_CS50x_India_Secondary
결과절: MITx_6.00x_India_Secondary
이며,
복합평가지표는 0.0113375 이다.

생성된 규칙을 plot 함수의 "graph" method 을 통해서 시각화하였다.



내부에 생성된 원은 각 아이템끼리의 연관규칙을 나타낸다. 원의 크기는 support 에 비례하고, 원의 농도 (진한 정도) 는 lift를 나타낸다.

```
#support x confidence x lift
fixed_h_df <- DATAFRAME(fixed_h)
fixed_h_df$Perf_New <- fixed_h_df$support * fixed_h_df$confidence * fixed_h_df$lift
fixed_h_df <- fixed_h_df[order(fixed_h_df[,8],decreasing = T),]
fixed_h_df

plot(fixed_h, method = "graph")
rules <- subset(fixed_h, lhs %pin% c("HarvardX_CS50x_UnitedStates_Master's"))
inspect(rules)
rules1 <- subset(fixed_h, lhs %pin% c("MITx_6.00x_UnitedStates_Master's"))
inspect(rules1)

rules2 <- subset(fixed_h, lhs %pin% c("MITx_6.00x_UnitedStates_Secondary"))
inspect(rules2)
rules3 <- subset(fixed_h, lhs %pin% c("MITx_6.002x_UnitedStates_Secondary"))
inspect(rules3)

rules4 <- subset(fixed_h, lhs %pin% c("MITx_6.002x_India_Bachelor's"))
inspect(rules4)
rules5 <- subset(fixed_h, lhs %pin% c("MITx_6.00x_India_Bachelor's"))
inspect(rules5)
```

위의 plot 중 양방향을 가지고 있는 규칙 중 세가지를 선택 후 분석을 진행해보았다. 선정한 규칙의 지표들을 쉽게 찾기 위해 subset 변수와 그 안의 %pin% 명령어를 통해 해당 값을 도출하였다. 선정한 규칙은 다음과 같다:

조건절 1: HarvardX_CS50x_UnitedStates_Master's
결과절 1: MITx_6.00x_UnitedStates_Master's

조건절 2: MITx_6.00x_UnitedStates_Secondary
결과절 2: MITx_6.002x_UnitedStates_Secondary

조건절 3: MITx_6.002x_India_Bachelor's
결과절 3: MITx_6.00x_India_Bachelor's

	조건절	결과절	Support	Confidence	Lift
1	HarvardX_CS50x_UnitedStates_Master's	MITx_6.00x_UnitedStates_Master's	0.001218535	0.1698505	11.42946
	MITx_6.00x_UnitedStates_Master's	HarvardX_CS50x_UnitedStates_Master's	0.001218535	0.08199679	11.42946
2	MITx_6.00x_UnitedStates_Secondary	MITx_6.002x_UnitedStates_Secondary	0.001939526	0.07363420	10.703875
	MITx_6.002x_UnitedStates_Secondary	MITx_6.00x_UnitedStates_Secondary	0.001939526	0.2819402	10.70387
3	MITx_6.002x_India_Bachelor's	MITx_6.00x_India_Bachelor's	0.003092516	0.1359885	5.842109
	MITx_6.00x_India_Bachelor's	MITx_6.002x_India_Bachelor's	0.003092516	0.13285550	5.842109

위의 표에서 각 규칙에 대한 support, confidence, lift 를 확인 해 볼 수 있다. 표를 확인해 보면, 양방향을 가지고 있는 규칙 (아이템셋) 은 조건절과 결과절의 위치를 바꿔도 support 값과 lift 값이 동일한 것을 알 수 있다. 하지만 confidence 값은 같지 않았다.

$$\text{confidence}(A \rightarrow B) = \frac{P(A,B)}{P(A)}$$

이는 confidence 를 구하는 수식에서 이유를 확인 할 수 있다. 수식을 조금 더 쉽게 설명하기 위해 A: 조건절, B: 결과절, P(A): 조건절의 support, P(A,B): 규칙의 support 로 명명하겠다.

각 규칙에 대한 confidence 를 계산할 때, 조건절의 support (분모) 가 달라지기 때문에, confidence 가 바뀐다. 조건절의 support (발생할 확률)이 커지면 confidence 값이 작아지게 되는데, 이를 통해 각 규칙의 조건절이 일어날 확률을 정성적으로 알 수 있다. 예를 들어, 첫번째 규칙 (A → B) 을 보면 confidence 가 0.1698505 이고, 이에 상응하는 반대 규칙 (B → A) 의 confidence 는 0.08199679 이다. 이를 통해 알 수 있는 사실은, 반대 규칙의 조건절 (B)이 일어날 확률이 조건절 (A) 보다 일어날 확률, 즉 support P(A) 보다 크기 때문에 반대 규칙 (B → A) 의 confidence 가 작아진 것을 알 수 있다.

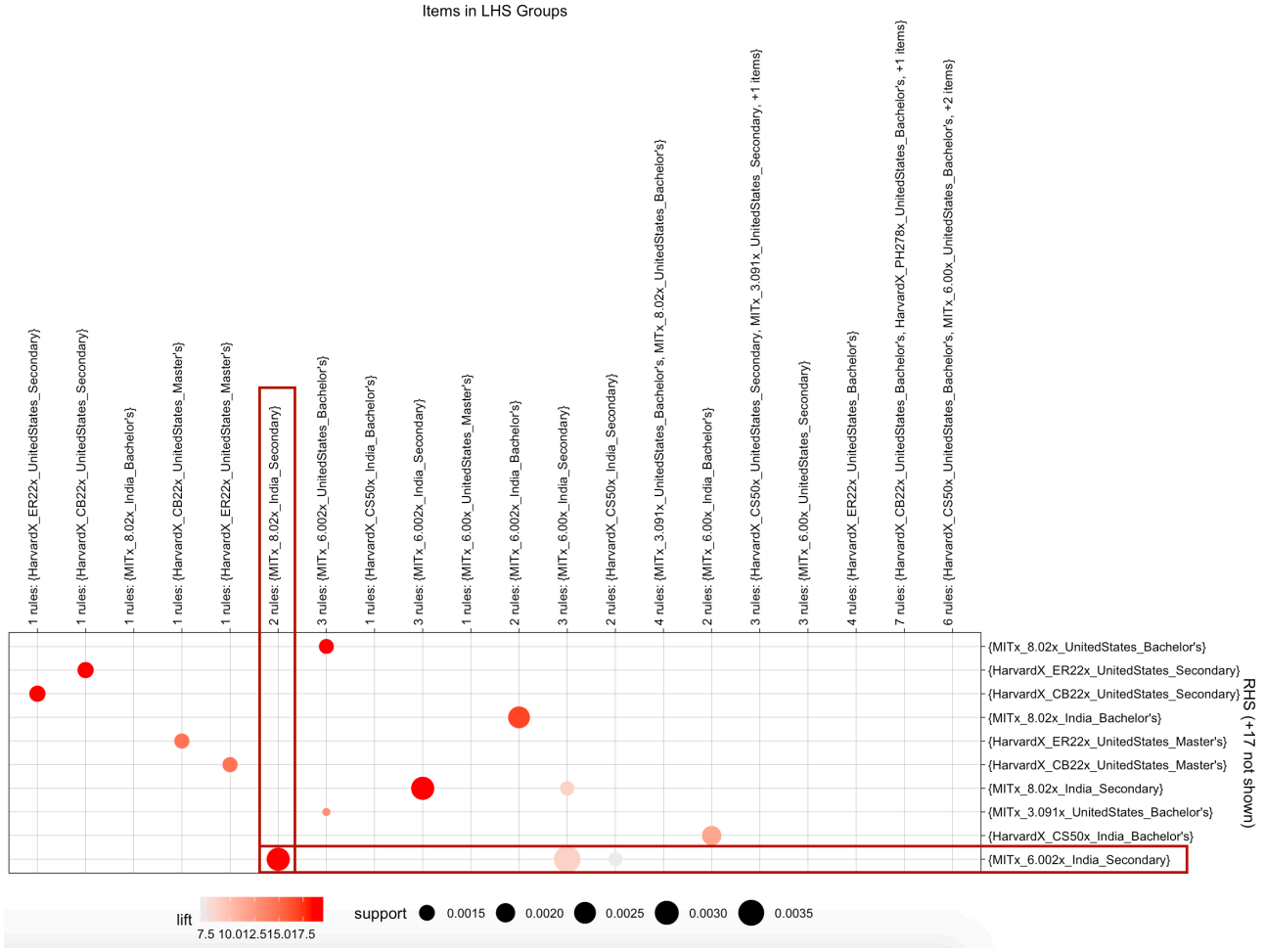
∴ Since P(B) > P(A), confidence (A → B) > confidence (B → A).

[Extra Question] Additional Visualization

추가적인 연관규칙분석의 시각화를 위해 먼저 arulesViz 패키지에서 제공하는 메서드들을 살펴보았다.

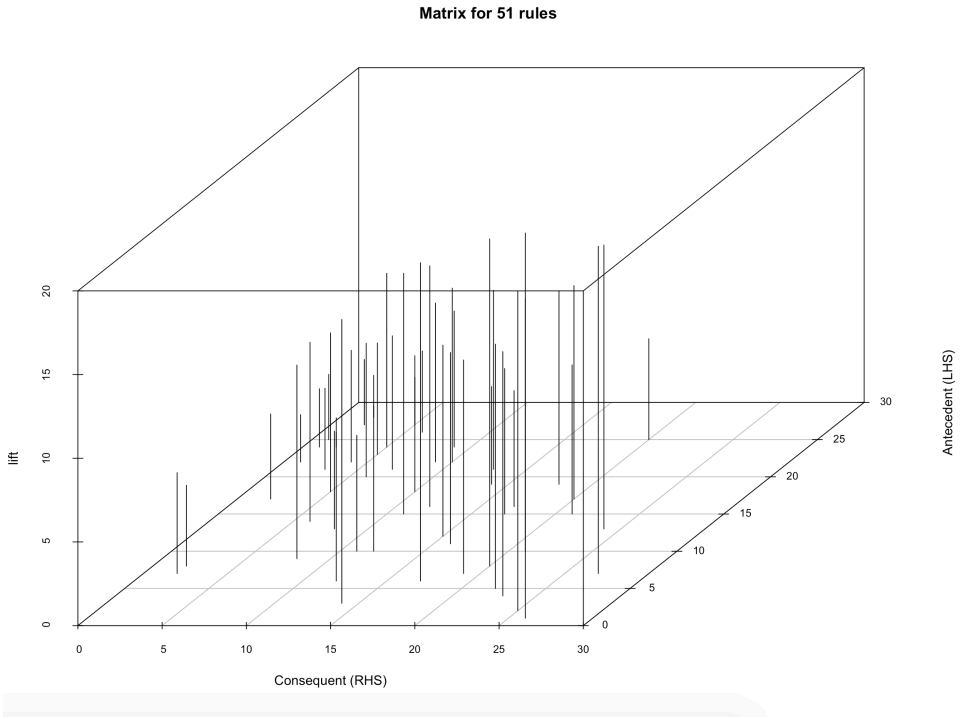
```
#Extra Question
plot(fixed_h, method = "grouped")
plot(fixed_h, method = "matrix", engine = "3d", measure = "lift")
plot(fixed_h, measure = c("support", "lift"), shading = "confidence", jitter = 0)
plot(fixed_h, method = "two-key plot")
plot(fixed_h, method = "paracoord")
```

제공되는 메서드들은 다양했지만, 시각화와 이에 대한 정성적 해석이 제일 직관적인 method = “grouped” 를 이용하여 분석을 진행해보았다.



Method = “grouped” 를 사용한 plot 이다. 연관규칙의 조건절 (LHS) 와 결과절 (RHS) 을 기준으로 결과를 보여준다. 도식된 원의 크기는 legend 의 기준에 따른 support 척도를 보여주며, 원의 색상은 legend 의 기준에 따라 lift 를 보여준다. 모니터의 크기가 그다지 크지 않기 때문에 결과절의 17개의 아이템이 생략된 것을 알 수 있지만, 제일 연관규칙이 또렷한 결과절을 도식하기 때문에 충분한 해석이 가능하다고 판단하였다.

조건절 앞에 있는 숫자는 해당 조건으로 시작하는 연관규칙의 개수를 나타낸다. 예를 들어서 MITx_8.02x_India_Secondary 조건절과 MITx_6.002x_India_Secondary 의 결과절을 갖는 규칙은 총 2개임을 알 수 있다. 또한 이 규칙에 대한 correlating 원의 색을 보면, lift 가 다른 규칙들 보다 높은 것을 알 수 있다. 각 지표들의 정확한 값을 명시하는 legend 가 없더라도 원의 크기와 색상을 통해 정성적인 해석이 가능하다.



위의 3d 그래프는 method = “matrix”, engine = “3d” 로 설정하여 도식된 그래프이다.

```
Itemsets in Antecedent (LHS)
 [1] "{HarvardX_CB22x_UnitedStates_Secondary}"
 [2] "{HarvardX_ER22x_UnitedStates_Secondary}"
 [3] "{MITx_8.02x_India_Bachelor's}"
 [4] "{HarvardX_CB22x_UnitedStates_Master's}"
 [5] "{HarvardX_ER22x_UnitedStates_Master's}"
```

```
Itemsets in Consequent (RHS)
 [1] "{HarvardX_CS50x_UnitedStates_Bachelor's}"
 [2] "{MITx_6.00x_UnitedStates_Bachelor's}"
 [3] "{HarvardX_CS50x_UnitedStates_Secondary}"
 [4] "{MITx_14.73x_UnitedStates_Bachelor's}"
 [5] "{HarvardX_ER22x_UnitedStates_Bachelor's}"
```

출력창에 나타나는 각 조건절과 결과절의 숫자 지표를 통해 해석이 가능하며, 상응하는 bar 은 각 규칙에 대한 lift 의 상대적 척도이다. 3차원으로 도식한다는 장점이 있지만, Method = “grouped” 보다 직관적인 해석력이 미미하므로 주 분석으로 사용하지 않았다.

이후 코드에서 볼 수 있듯이 살펴본 메서드들 (two - key plot, paracoord) 은 추가적인 insight 를 주지 않는다고 판단하여 제외했다.