

11장 연관규칙



충북대학교 정보통계학과 나종화
(cherin@cbnu.ac.kr)

CONTENTS

11.1 서론

11.2 주요 개념

11.3 연관규칙 알고리즘

11.1 서론

- 연관규칙(association rule) 학습은 대형 데이터베이스에서 변수 간의 흥미로운 관계를 발견하기 위한 규칙-기반 기계 학습 방법이다. 이것은 흥미로운 측도를 사용하여 데이터베이스에서 발견된 강력한 규칙을 식별하기 위한 것이다.
- Agrawal 등(1993)은 강력한 규칙의 개념을 바탕으로 슈퍼마켓 POS(point-of-sale) 시스템에서 기록한 대규모 거래 데이터에서 제품 간의 규칙성을 발견하는 연관 규칙을 소개하였다. 예를 들어, 슈퍼마켓의 판매 데이터에서 발견된 “{양파, 감자} ⇒ {버거}” 규칙은 고객이 양파와 감자를 함께 구매하면 햄버거 고기도 사기 쉽다는 것을 알 수 있다.

11.1 서론

- 이러한 정보는 예를 들어, 프로모션 가격 또는 제품 배치와 같은 마케팅 활동에 관한 결정을 위한 기초 자료로 사용될 수 있다.
- 장바구니 분석(market basket analysis)에 대한 위의 예제 외에도 웹 사용 마이닝, 침입 탐지, 연속 생산 및 생물 정보학을 비롯한 많은 분야에서 연관규칙이 사용되고 있다.
- 순차연관성 마이닝(sequence mining)과는 달리, 연관규칙 학습은 일반적으로 트랜잭션 내에서 또는 트랜잭션 전반에서 항목의 순서는 고려하지 않는다.

11.2 주요 개념

- 연관규칙은 $X \Rightarrow Y$ 로 표시한다. 예를 들어, 판매 제품 간의 연관규칙이 $\{\text{onion, potato}\} \Rightarrow \{\text{meat}\}$ 이면, $\{\text{onion, potato}\}$ 를 구매하면 meat 도 구매하는 규칙으로 해석 할 수 있다.
- 의미 있는 연관규칙의 선택을 위해 다음의 측도가 유용하게 사용된다. 상품에 대한 연관규칙 $X \Rightarrow Y$ 에 대해, 지지도, 신뢰도, 향상도는 다음과 같이 정의된다.
 - 지지도(support)는 전체 구매 건수 가운데 상품 X 와 Y 를 동시에 구매한 비율을 의미하며 $P(X \cup Y)$ 으로 나타낸다.
 - 신뢰도(confidence)는 상품 X 를 구매한 건수 가운데 Y 도 같이 구매한 비율을 의미하며 조건부 확률 $P(Y|X)$ 로 나타낸다.
 - 향상도(lift)는 전체에서 상품 Y 를 구매한 비율에 비해 X 를 구매한 고객이 Y 를 구매한 비율이 몇 배 인가를 나타내며 $P(Y|X)/P(Y)$ 로 나타낸다.

11.2 주요 개념

- 이를 정리하면 다음과 같다.
- **지지도**(support) = $P(X \cup Y)$
- **신뢰도**(confidence) = $P(Y|X) = \frac{P(X \cup Y)}{P(X)} = \frac{\text{support}}{P(X)}$
- **향상도**(lift) = $\frac{P(Y|X)}{P(Y)} = \frac{P(X \cup Y)}{P(X)P(Y)} = \frac{\text{confidence}}{P(Y)}$

11.3 연관규칙 알고리즘

- 연관규칙을 생성하는 알고리즘은 다양하다. 이 가운데 Apriori, Eclat 및 FP-Growth 알고리즘은 잘 알려진 알고리즘이지만 빈발항목집합(frequent itemsets)을 마이닝하는 알고리즘이기 때문에 절반 만 수행한다.
- 다음 단계는 데이터베이스에서 발견된 빈발항목집합에서 규칙을 생성 하는 것이다.

(a) Apriori 알고리즘

- 간단한 예를 통해 Apriori 알고리즘을 소개한다. 다음의 [표 11.1]을 슈퍼마켓에서의 거래 자료 (transaction data)를 생각하자.

11.3 연관규칙 알고리즘

(a) Apriori 알고리즘

- 간단한 예를 통해 Apriori 알고리즘을 소개한다. 다음의 [표 11.1]을 슈퍼마켓에서의 거래 자료 (transaction data)를 생각하자.

[표 11.1] 슈퍼마켓의 거래 항목 자료

항목 집합
{a,b,c,d}
{a,b,d}
{a,b}
{b,c,d}
{b,c}
{c,d}
{b,d}

- 저장된 위의 자료로부터, Apriori를 사용하여 빈발항목 (frequent item) 집합을 결정하고자 한다. 이를 위해 최소 3건의 거래가 일어난 항목 집합을 빈발항목으로 정하자(즉, 발생 수의 임계값을 3으로 정함).

11.3 연관규칙 알고리즘

[단계 1] 데이터베이스로부터 각 원소 항목의 지지도¹⁾를 구한다([표 11.2]).

[표 11.2] 1항목 집합의 지지도

항목	지지도
{a}	3
{b}	6
{c}	4
{d}	5

- 위 목록에서 크기 1의 모든 항목집합은 적어도 지지도 3을 가지므로 빈발항목이 된다.
- 1) 지지도는 원래 (해당 항목의 거래 건수)/(총 거래 건수)이나, 여기서는 편의상 분자만을 지지도라 부르는 기로 한다.

11.3 연관규칙 알고리즘

[단계 2] 빈발항목의 모든 쌍의 목록을 생성하고(2항목 후보 빈발항목 집합), 지지도를 구한다([표 11.3]).

[표 11.3] 2항목 후보 빈발항목 집합과 지지도

항목	지지도
{a,b}	3
{a,c}	1
{a,d}	2
{b,c}	3
{b,d}	4
{c,d}	3

- 위의 표에서 지지도 기준(3 이상)을 만족하는 쌍 즉, 빈발항목 집합은 {a,b}, {b,c}, {b,d}와 {c,d}이다.
 - 쌍 {a,c}와 {a,d}는 비빈발항목 집합에 속하므로, 이를 포함하는 더 큰 항목집합은 빈발항목이 될 수 없다. 이 방식으로 집합에 대한 가지치기를 수행할 수 있다.
- (Apriori 규칙 2) 한 항목집합이 비빈발하다면 이 항목집합을 포함하는 모든 집합은 비빈발 항목집합이다.

11.3 연관규칙 알고리즘

[단계 3] 2항목 빈발항목집합간의 조합을 이용하여 3항목 후보 빈발항목 집합의 목록을 생성한다.
이 과정에서 (Apriori 규칙 1)이 적용되어 가지치기가 수행된다(항목 {a,b,c}은 제거됨).

[표 11.4] 3항목 후보 빈발항목 집합과 지지도

항목	지지도
{b,c,d}	2

- 위의 3원소 집합은 지지도 기준을 만족하지 못하므로 빈발항목집합이 아니다. 알고리즘은 여기서 중단된다.
- (Apriori 규칙 1) 한 항목집합이 빈발하다면 이 항목집합의 모든 부분집합은 역시 빈발항목집합입니다.

11.3 연관규칙 알고리즘

(b) FP-성장 알고리즘

- FP-Growth 알고리즘은 후보 빈발항목집합 생성 없이 FP-Tree(Frequent Pattern Tree)를 이용하여 빈발항목집합을 추출하기 위해 고안된 방법이다. 불필요한 후보 빈발항목집합을 생성하지 않음으로써, Apriori 알고리즘보다 빠르게 빈발항목집합을 추출할 수 있다.
- 이 방법은 FP-Tree를 생성하는 단계와 FP-Tree로부터 빈발항목집합을 추출하는 단계, 총 두 가지 단계로 이루어져있다. 자세한 설명은 생략한다.

11.3 연관규칙 알고리즘

예제 1 Titanic 자료에 대해 연관분석을 수행한다. 이 자료는 승객 2201명에 대한 객실 등급, 성별, 연령, 생존 여부를 포함한다.

```
> # Titanic 자료(테이블 객체)를 분석용 자료로 변환
```

```
> data(Titanic)
```

```
> titan.df <- as.data.frame(Titanic)
```

```
> head(titan.df)
```

	Class	Sex	Age	Survived	Freq
1	1st	Male	Child	No	0
2	2nd	Male	Child	No	0
3	3rd	Male	Child	No	35
4	Crew	Male	Child	No	0
5	1st	Female	Child	No	0
6	2nd	Female	Child	No	0

11.3 연관규칙 알고리즘

```
> summary(titanic)
```

Class	Sex	Age	Survived
1st :325	Female: 470	Adult:2092	No :1490
2nd :285	Male :1731	Child: 109	Yes: 711
3rd :706			
Crew:885			

```
> titanic <- NULL
```

```
> for(i in 1:4) { titanic <- cbind(titanic,  
                                rep(as.character(titan.df[,i]), titan.df$Freq)) }
```

```
> titanic <- as.data.frame(titanic)
```

```
> names(titanic) <- names(titanic.df)[1:4]
```

11.3 연관규칙 알고리즘

> **titanic**

	Class	Sex	Age	Survived
1	3rd	Male	Child	No
2	3rd	Male	Child	No
3	3rd	Male	Child	No
...				
2200	Crew	Female	Adult	Yes
2201	Crew	Female	Adult	Yes

11.3 연관규칙 알고리즘

```
> ## 연관규칙 분석: apriori 알고리즘으로 연관 규칙 찾기
> # apriori{arules} 함수 이용
> library(arules)
> # 모든 규칙 생성
> rules.all <- apriori(titanic)
```

Apriori

Parameter specification:

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

... (생략)

11.3 연관규칙 알고리즘

```
> options(digits=3)
```

```
> inspect(rules.all)
```

	lhs	rhs	support	confidence	lift
[1]	{}	=> {Age=Adult}	0.950	0.950	1.000
[2]	{Class=2nd}	=> {Age=Adult}	0.119	0.916	0.964
[3]	{Class=1st}	=> {Age=Adult}	0.145	0.982	1.033
[4]	{Sex=Female}	=> {Age=Adult}	0.193	0.904	0.951
[5]	{Class=3rd}	=> {Age=Adult}	0.285	0.888	0.934
[6]	{Survived=Yes}	=> {Age=Adult}	0.297	0.920	0.968
[7]	{Class=Crew}	=> {Sex=Male}	0.392	0.974	1.238
[8]	{Class=Crew}	=> {Age=Adult}	0.402	1.000	1.052
[9]	{Survived=No}	=> {Sex=Male}	0.620	0.915	1.164
[10]	{Survived=No}	=> {Age=Adult}	0.653	0.965	1.015
...					
[26]	{Class=Crew,Sex=Male,Survived=No}	=> {Age=Adult}	0.304	1.000	1.052
[27]	{Class=Crew,Age=Adult,Survived=No}	=> {Sex=Male}	0.304	0.996	1.266

11.3 연관규칙 알고리즘

```
> # 규칙의 우변(rhs)가 생존 여부(Survived)와 관계된 규칙
> # 설정값 변경: 최소부분집합크기=2, 최소지지도=0.005, 최소신뢰도=0.8
> rules <- apriori(titanic, control = list(verbose=F),
  parameter = list(minlen=2, supp=0.005, conf=0.8),
  appearance = list(rhs=c("Survived=No", "Survived=Yes"),
  default="lhs"))
> # 향상도(lift) 기준으로 정렬
> rules.sorted <- sort(rules, by="lift")
```

11.3 연관규칙 알고리즘

> # 규칙 확인

> inspect(rules.sorted)

	lhs	rhs	support	confidence	lift
[1]	{Class=2nd, Age=Child}	=> {Survived=Yes}	0.01090	1.000	3.10
[2]	{Class=2nd, Sex=Female, Age=Child}	=> {Survived=Yes}	0.00591	1.000	3.10
[3]	{Class=1st, Sex=Female}	=> {Survived=Yes}	0.06406	0.972	3.01
[4]	{Class=1st, Sex=Female, Age=Adult}	=> {Survived=Yes}	0.06361	0.972	3.01
[5]	{Class=2nd, Sex=Female}	=> {Survived=Yes}	0.04225	0.877	2.72
[6]	{Class=Crew, Sex=Female}	=> {Survived=Yes}	0.00909	0.870	2.69
[7]	{Class=Crew, Sex=Female, Age=Adult}	=> {Survived=Yes}	0.00909	0.870	2.69
[8]	{Class=2nd, Sex=Female, Age=Adult}	=> {Survived=Yes}	0.03635	0.860	2.66
[9]	{Class=2nd, Sex=Male, Age=Adult}	=> {Survived=No}	0.06997	0.917	1.35
[10]	{Class=2nd, Sex=Male}	=> {Survived=No}	0.06997	0.860	1.27
[11]	{Class=3rd, Sex=Male, Age=Adult}	=> {Survived=No}	0.17583	0.838	1.24
[12]	{Class=3rd, Sex=Male}	=> {Survived=No}	0.19173	0.827	1.22

11.3 연관규칙 알고리즘

```
> # 중복되는 규칙 찾기
> subset.matrix <- is.subset(rules.sorted, rules.sorted)
> subset.matrix[lower.tri(subset.matrix, diag=T)] <- NA
> redundant <- colSums(subset.matrix, na.rm = T) >= 1
> which(redundant)
{Class=2nd,Sex=Female,Age=Child,Survived=Yes}
      2
{Class=1st,Sex=Female,Age=Adult,Survived=Yes}
      4
{Class=Crew,Sex=Female,Age=Adult,Survived=Yes}
      7
{Class=2nd,Sex=Female,Age=Adult,Survived=Yes}
      8
```

11.3 연관규칙 알고리즘

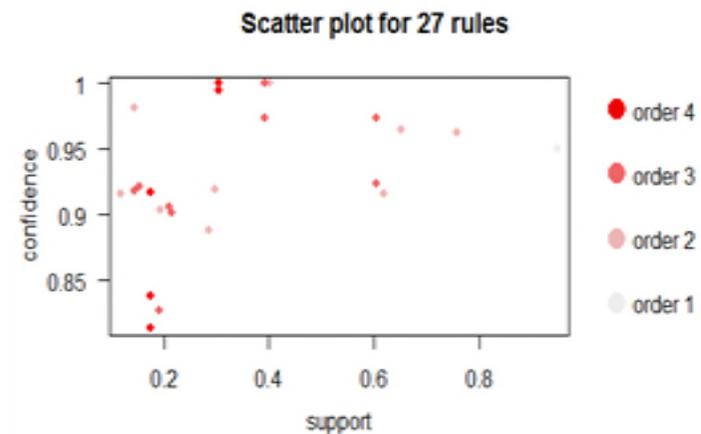
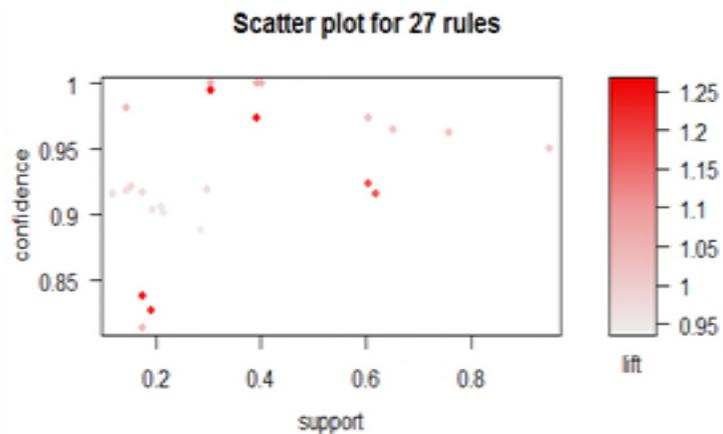
> # 중복되는 규칙 삭제

> rules.pruned <- rules.sorted[!redundant]

	lhs	rhs	support	confidence	lift
[1]	{Class=2nd, Age=Child}	=> {Survived=Yes}	0.01090	1.000	3.10
[2]	{Class=1st, Sex=Female}	=> {Survived=Yes}	0.06406	0.972	3.01
[3]	{Class=2nd, Sex=Female}	=> {Survived=Yes}	0.04225	0.877	2.72
[4]	{Class=Crew, Sex=Female}	=> {Survived=Yes}	0.00909	0.870	2.69
[5]	{Class=2nd, Sex=Male, Age=Adult}	=> {Survived=No}	0.06997	0.917	1.35
[6]	{Class=2nd, Sex=Male}	=> {Survived=No}	0.06997	0.860	1.27
[7]	{Class=3rd, Sex=Male, Age=Adult}	=> {Survived=No}	0.17583	0.838	1.24
[8]	{Class=3rd, Sex=Male}	=> {Survived=No}	0.19173	0.827	1.22

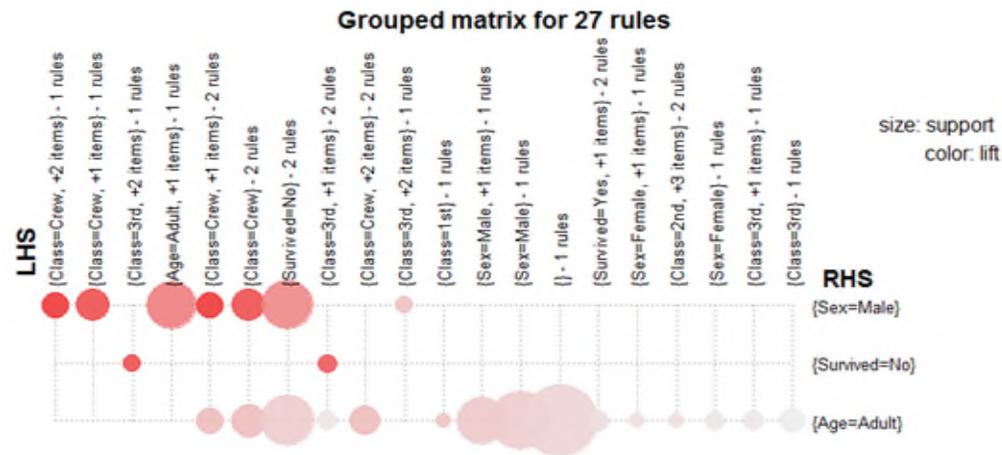
11.3 연관규칙 알고리즘

```
> ## 연관규칙 시각화
> library(arulesViz)
> plot(rules.all)      # 디폴트 옵션: measure=c("support",
"confidence"), shading="lift"
> plot(rules.all, shading="order") # 규칙번호에 따라 음영부여
```



11.3 연관규칙 알고리즘

```
> plot(rules.all, method="grouped")
```



해석

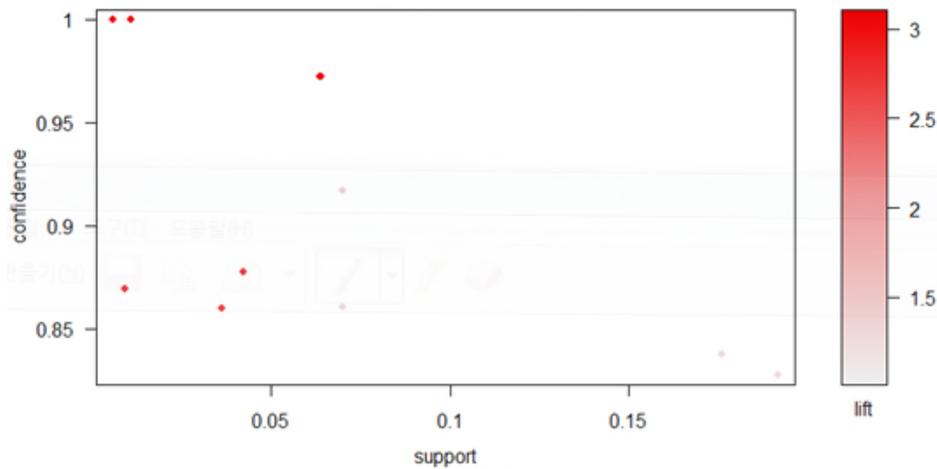
좌측상단의 첫 번째 원에 대한 해석은 다음과 같다. {Class=Crew, +2 items}-1 rules 은 연관규칙의 좌변(LHS)이 “[Class=Crew, +2 items]: {승무원석+ 2개 조건이 추가}”임을 말하며, 규칙의 우변(RHS)은 {Sex=Male}임을 의미하며, 이 조건을 만족하는 연관규칙이 1개(1 rules) 있음을 나타낸다.

원의 크기는 지지도를 나타내며, 색이 진할수록 향상도가 큼을 의미한다.

11.3 연관규칙 알고리즘

```
> plot(rules.sorted) # 12개 규칙
> plot(rules.sorted, method="grouped")
```

Scatter plot for 12 rules

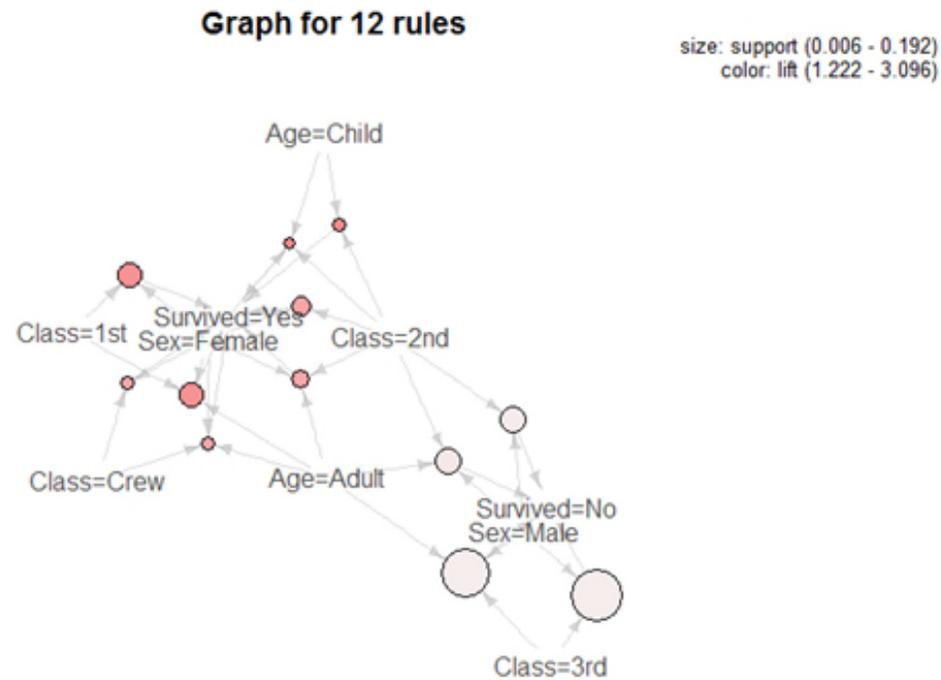


Grouped matrix for 12 rules



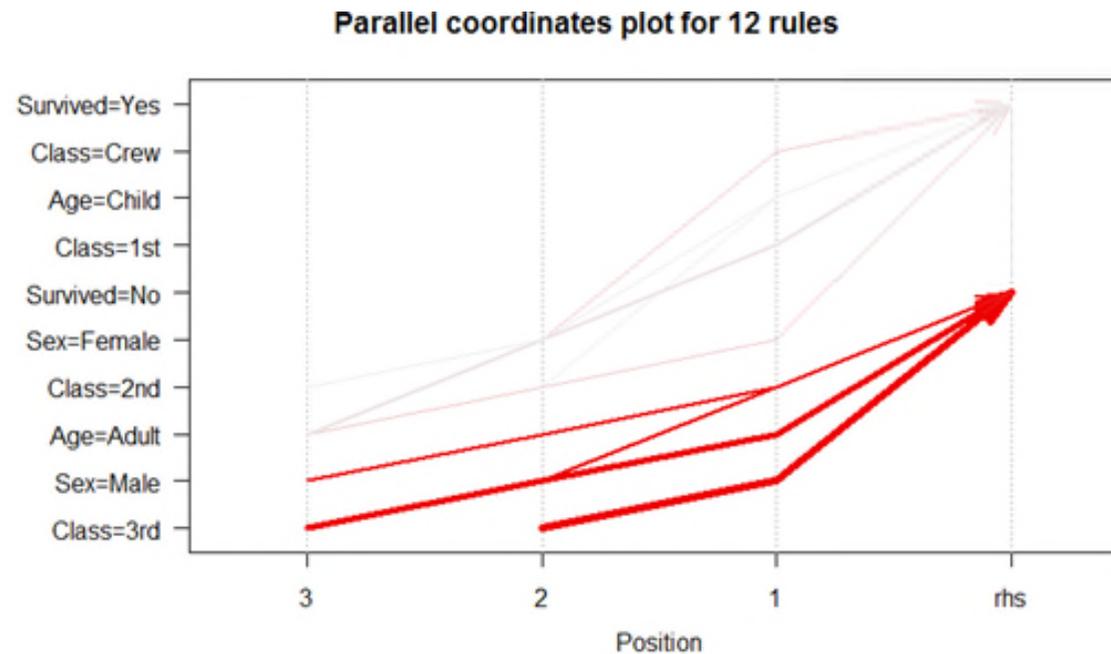
11.3 연관규칙 알고리즘

```
> plot(rules.sorted, method="graph", control=list(type="items"))  
> # 10개 item(10=4+2+2+2)
```



11.3 연관규칙 알고리즘

- > # 평행좌표그림
- > `plot(rules.sorted, method="paracoord", control=list(reorder=TRUE))`
- > # 12개 규칙: 규칙 [1]~[8]은 Survived=Yes, [9]~[12]는 Survived=No



11.3 연관규칙 알고리즘

```
> ## 대화식(interactive) 그림
> # 선택된 규칙을 조사(inspect), 줌인(줌아웃), 규칙 필터링(color key에서 절단값 클릭)
> plot(rules.sorted, measure=c("support", "lift"),
       shading="confidence", interactive=TRUE) # 동적 시각화 제공
```

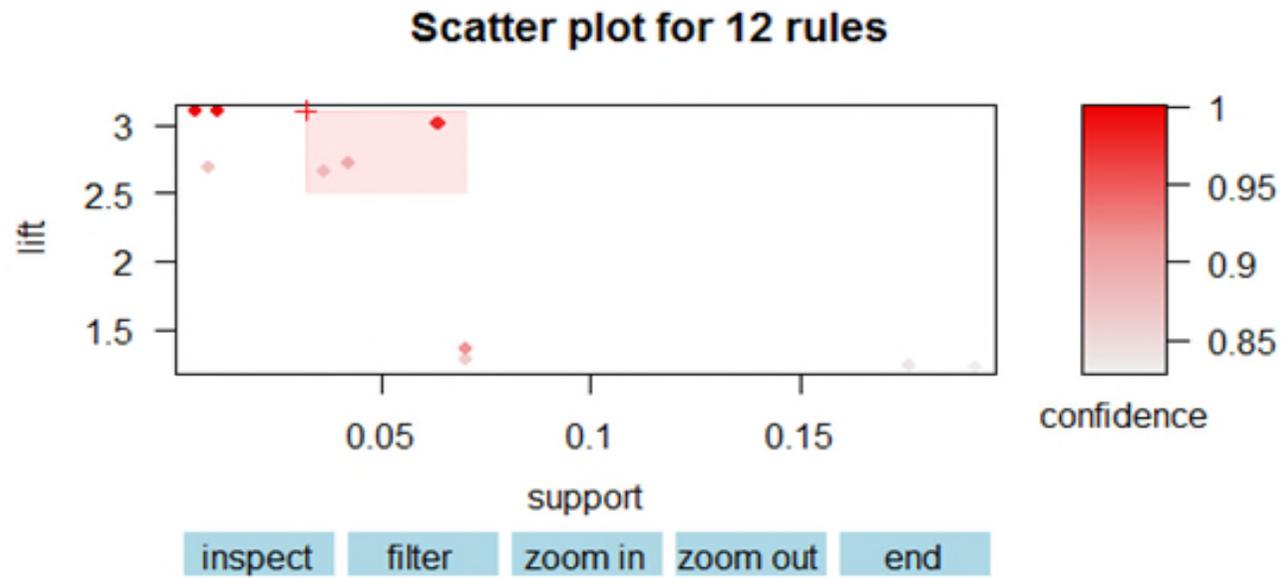
Interactive mode.

Select a region with two clicks!

Number of rules selected: 4

	lhs	rhs	support	confidence	lift	order
[1]	{Class=1st,Sex=Female}	=> {Survived=Yes}	0.0641	0.972	3.01	3
[2]	{Class=1st,Sex=Female,Age=Adult}	=> {Survived=Yes}	0.0636	0.972	3.01	4
[3]	{Class=2nd,Sex=Female}	=> {Survived=Yes}	0.0423	0.877	2.72	3
[4]	{Class=2nd,Sex=Female,Age=Adult}	=> {Survived=Yes}	0.0363	0.860	2.66	4

11.3 연관규칙 알고리즘



11.3 연관규칙 알고리즘

> ## 행렬-기반 시각화

> plot(rules.sorted, method="matrix", measure="lift")

Itemsets in Antecedent (LHS)

[1] "{Class=2nd, Age=Child}"

[2] "{Class=2nd, Sex=Female, Age=Child}"

[3] "{Class=1st, Sex=Female}"

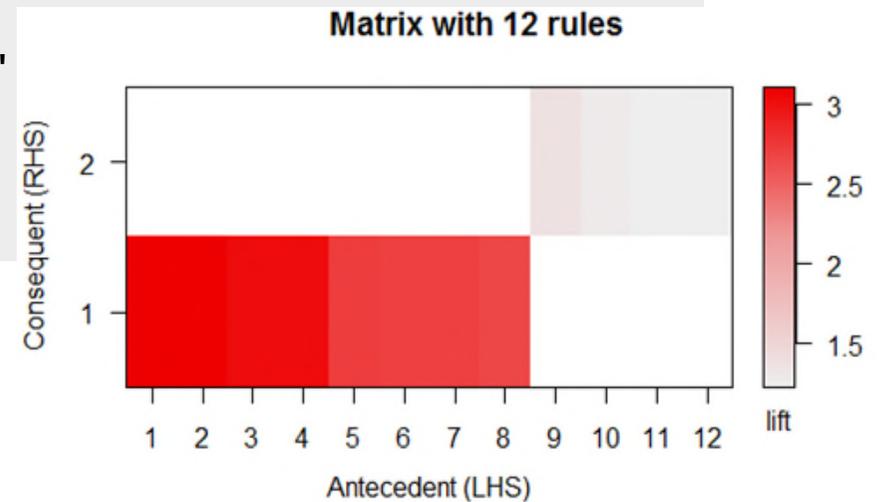
...

[11] "{Class=3rd, Sex=Male, Age=Adult}"

[12] "{Class=3rd, Sex=Male}"

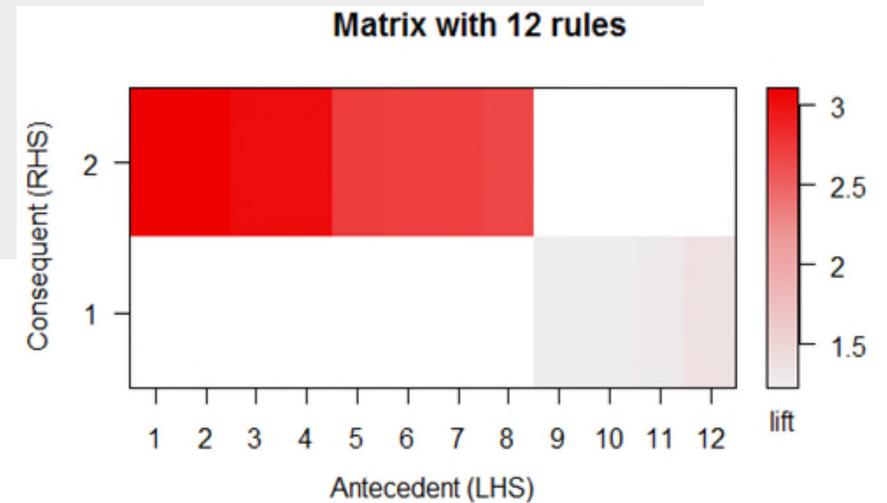
Itemsets in Consequent (RHS)

[1] "{Survived=Yes}" "{Survived=No}"



11.3 연관규칙 알고리즘

```
> plot(rules.sorted, method="matrix", measure="lift",
control=list(reorder=TRUE))
Itemsets in Antecedent (LHS)
[1] "{Class=2nd, Age=Child}"
[2] "{Class=2nd, Sex=Female, Age=Child}"
[3] "{Class=1st, Sex=Female}"
...
[11] "{Class=2nd, Sex=Male}"
[12] "{Class=2nd, Sex=Male, Age=Adult}"
Itemsets in Consequent (RHS)
[1] "{Survived=Yes}" "{Survived=No}"
```



11.3 연관규칙 알고리즘

```
> plot(rules.sorted, method="matrix", measure=c("lift",  
"confidence"),  
       control=list(reorder=TRUE))
```

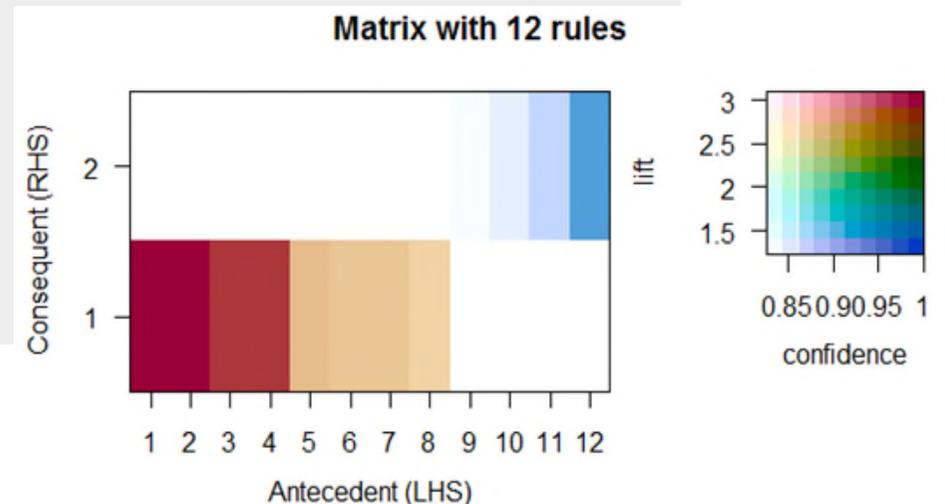
Itemsets in Antecedent (LHS)

```
[1] "{Class=2nd, Age=Child}"  
[2] "{Class=2nd, Sex=Female, Age=Child}"  
[3] "{Class=1st, Sex=Female}"  
...
```

```
[11] "{Class=2nd, Sex=Male}"  
[12] "{Class=2nd, Sex=Male, Age=Adult}"
```

Itemsets in Consequent (RHS)

```
[1] "{Survived=Yes}" "{Survived=No}"
```



11.3 연관규칙 알고리즘

```
> plot(rules.sorted, method="matrix3D", measure="lift",  
       control=list(reorder=TRUE))
```

Itemsets in Antecedent (LHS)

```
[1] "{Class=2nd, Age=Child}"
```

```
[2] "{Class=2nd, Sex=Female, Age=Child}"
```

```
[3] "{Class=1st, Sex=Female}"
```

```
...
```

```
[11] "{Class=2nd, Sex=Male}"
```

```
[12] "{Class=2nd, Sex=Male, Age=Adult}"
```

Itemsets in Consequent (RHS)

```
[1] "{Survived=No}" "{Survived=Yes}"
```

