DATA 624
Association Rules

Joby John
Mike Gankhuyag
Albina Gallyavova

Table of Contents

Association Rules

- Rule-based machine learning method for discovering interesting relations between variables in large databases
- If-then statements that help to show the probability of relationships between data items
  - A → B[Support,Confidence]
    - Computer -> Anti-virus Software[Support=20%,confidence=60%]
  - {Diaper} → {Beer}
- Marketing, **Basket Data Analysis (or Market Basket Analysis)** in retailing, bioinformatics, medical diagnosis, Web mining, and scientific data analysis
- Analysis of credit card purchases, Identification of fraudulent medical insurance claims

Market Basket Analysis

- Uncovers associations between products by looking for combinations of products that frequently co-occur in transactions
- Allows retailers to identify relationships between the items that people buy
- Association Rules are widely used to analyze retail basket or transaction data, and to identify strong rules in transaction data using measures of interestingness

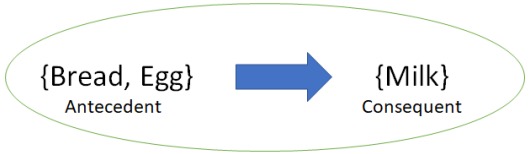
Market Basket Analysis cont’d

Each row in this table corresponds to a transaction, which contains a uniqueidentifier labeled TID

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

{Diaper} → {Beer}
{Milk, Bread} → {Diaper}
{Beer, Bread} → {Milk}

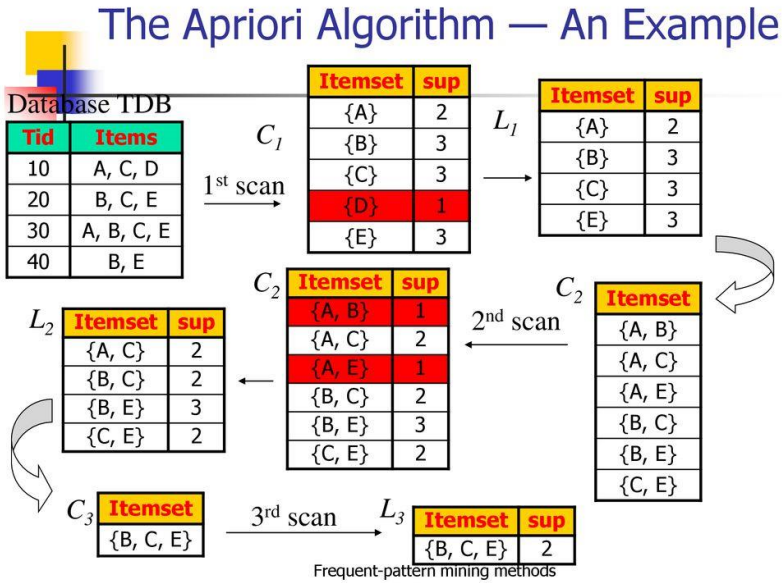
- Itemset
  - A collection of one or more items
    - Example: {Milk, Bread, Diaper}



Itemset = {Bread, Egg, Milk}

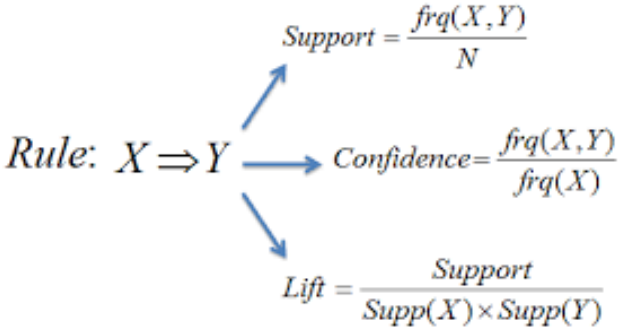
Apriori Algorithm

- Named Apriori because it uses prior knowledge of frequent itemset properties
- An iterative algorithm which looks for so-called frequent itemsets, which are representatives of sets of items that occur together in transactions
- Lists are notated with ‘L’ and ‘C’, which stands for ‘Large Itemset’ and ‘Candidate Itemset’
- It is assumed that the support of a frequent itemset is equal to or greater than a certain minimum support.
- Frequent itemsets are used to create association rules whose confidence is greater than or equal to a predefined minimum



Apriori Measurements

- Apriori primarily uses 3 measurements to generate association rules
  - **Support:** Measures how popular an itemset is
  - **Confidence:** Measures how likely item Y is purchased when item X is also purchased
  - **Lift:** Measures how likely item Y is purchased when item X is purchased, while controlling for how popular item Y is
- Support is used to generate frequent itemsets
- Confidence & Lift are used to generate association rules



Support

$$Support(\{X\} \rightarrow \{Y\}) = \frac{Transactions\ containing\ both\ X\ and\ Y}{Total\ number\ of\ transactions}$$

- Indicates the proportion of transactions in the dataset of all transactions containing
- Items are a candidate if they meet the support threshold
- Large itemsets are cross joined to generate association rules
- Example:
  - 10 transactions, 5 products
  - Minimum Support Threshold = 30%
  - Minimum Confidence Threshold = 60%

Transactions

Basket	Product 1	Product 2	Product 3
1	Peanut Butter	Jelly	Bread
2	Bananas	Apples	
3	Peanut Butter	Jelly	Bread
4	Apples	Bananas	
5	Peanut Butter	Jelly	
6	Bananas	Jelly	Apples
7	Peanut Butter	Bread	
8	Bread	Apples	Bananas
9	Bread	Peanut Butter	Jelly
10	Peanut Butter	Apples	Jelly

C1

Product	Number of Baskets	Support
Peanut Butter	6	60%
Apples	5	50%
Jelly	6	60%
Bananas	4	40%
Bread	5	50%

Support

$$Support(\{X\} \rightarrow \{Y\}) = \frac{Transactions\ containing\ both\ X\ and\ Y}{Total\ number\ of\ transactions}$$

- Generate 2nd & 3rd candidate by cross joining products and calculating support
- Continue to remove candidates that fail the minimum support count
- First step of Apriori mining is complete as we did not have a transaction with 4 products
- 5 itemsets satisfied the 30% Threshold

Frequent Itemsets

Itemset	Frequency	Support
{Peanut Butter, Jelly}	5	50%
{Peanut Butter, Bread}	4	40%
{Jelly, Bread}	3	30%
{Apples, Bananas}	4	40%
{Peanut Butter, Jelly, Bread}	3	30%

L2

Item 1	Item 2	Frequency	Support
Peanut Butter	Jelly	5	50%
Peanut Butter	Bread	4	40%
Peanut Butter	Apples	1	10%
Peanut Butter	Bananas	0	0%
Jelly	Bread	3	30%
Jelly	Apples	2	20%
Jelly	Bananas	1	10%
Bread	Apples	1	10%
Bread	Bananas	1	10%
Apples	Bananas	4	40%

C2

Itemset	Frequency	Support
{Peanut Butter, Jelly}	5	50%
{Peanut Butter, Bread}	4	40%
{Jelly, Bread}	3	30%
{Apples, Bananas}	4	40%

L3

Item 1	Item 2	Item 3	Frequency	Support
Peanut Butter	Jelly	Bread	3	30%
Peanut Butter	Jelly	Apples	1	10%
Peanut Butter	Jelly	Bananas	0	0%
Peanut Butter	Bread	Apples	0	0%
Peanut Butter	Bread	Bananas	0	0%
Jelly	Bread	Bananas	0	0%
Jelly	Bread	Apples	0	0%
Bananas	Apples	Bread	1	10%
Bananas	Apples	Jelly	1	10%

C3

Itemset	Frequency	Support
{Peanut Butter, Jelly, Bread}	3	30%

Confidence

$$Confidence(\{X\} \rightarrow \{Y\}) = \frac{Transactions\ containing\ both\ X\ and\ Y}{Transactions\ containing\ X}$$

- Conditional probability that given x is present, y will also be present
- Confidence metric is not symmetric or directed; for instance, the confidence for X->Y is different than the confidence for Y->X
- This is compared to the confidence threshold and eliminated if support does not meet threshold
- Item set is considered a strong rule if it satisfies the support and confidence thresholds
- Misrepresenting the importance of an association is a drawback

If {A,B,C,D} is a frequent itemset, candidate rules:

ABC →D,	ABD →C,	ACD →B,	BCD →A,
A →BCD,	B →ACD,	C →ABD,	D →ABC
AB →CD,	AC → BD,	AD → BC,	BC →AD,
BD →AC,	CD →AB,		

Frequent Itemsets

Itemset	Frequency	Support
{Peanut Butter, Jelly}	5	50%
{Peanut Butter, Bread}	4	40%
{Jelly, Bread}	3	30%
{Apples, Bananas}	4	40%
{Peanut Butter, Jelly, Bread}	3	30%

Candidate Rules

Itemset	Frequency	Support	Confidence
{Peanut Butter -> Jelly}	5	50%	83%
{Peanut Butter -> Bread}	4	40%	67%
{Jelly -> Bread}	3	30%	50%
{Apples -> Bananas}	4	40%	80%
{Jelly -> Peanut Butter}	5	50%	83%
{Bread -> Peanut Butter}	4	40%	80%
{Bread -> Jelly}	3	30%	60%
{Bananas -> Apples}	4	40%	100%
{Bread, Peanut Butter -> Jelly}	3	30%	75%
Jelly, Peanut Butter -> Bread}	3	30%	60%
{Bread, Jelly -> Peanut Butter}	3	30%	100%

11 possible rules were generated from the frequent itemsets
10 rules satisfied the minimum confidence threshold of 60%

Lift

$$Lift(\{X\} \rightarrow \{Y\}) = \frac{(Transactions\ containing\ both\ X\ and\ Y)/(Transactions\ containing\ X)}{Fraction\ of\ transactions\ containing\ Y}$$

- Lift is defined as the ratio of the confidence to the unconditional probability of the consequent B
- Measures the correlation between item sets in the rule. Correlation shows how item set X effects the item set Y
- Interpreting lift:
  - If the lift is greater than 1, then X & Y are dependent
  - If the lift is less than 1, then X will have negative effect on Y
  - If the lift is equal to 1, then X and Y are independent
- Rules that are considered dependent are “interesting”

Itemset	Frequency	Support	Confidence	Lift
{Peanut Butter -> Jelly}	5	50%	83%	1.39
{Peanut Butter -> Bread}	4	40%	67%	1.33
{Apples -> Bananas}	4	40%	80%	2.00
{Jelly -> Peanut Butter}	5	50%	83%	1.39
{Bread -> Peanut Butter}	4	40%	80%	1.33
{Bread -> Jelly}	3	30%	60%	1.00
{Bananas -> Apples}	4	40%	100%	2.00
{Bread, Peanut Butter -> Jelly}	3	30%	75%	1.25
{Jelly, Peanut Butter -> Bread}	3	30%	60%	1.20
{Bread, Jelly -> Peanut Butter}	3	30%	100%	1.67

9 rules were considered interesting as they had a lift greater than 1

MBA - R Implementation

- R includes apriori implementation in **arules** package
- Calls the C implementation of the Apriori algorithm by Christian Borgelt
- **arulesViz** allows to visualize rules

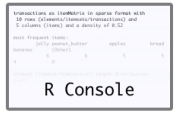
MBA - R Implementation

Data format

```
peanut_butter,jelly,bread
bananas,apples,
peanut_butter,jelly,bread
apples,bananas,
peanut_butter,jelly,
bananas,jelly,apples
peanut_butter,bread,
bread,apples,bananas
bread,peanut_butter,jelly
peanut_butter,apples,jelly
```

- read.transactions() to read data in
- summary()
  - most frequent items in the data set
  - transaction length distribution
  - extended transaction info

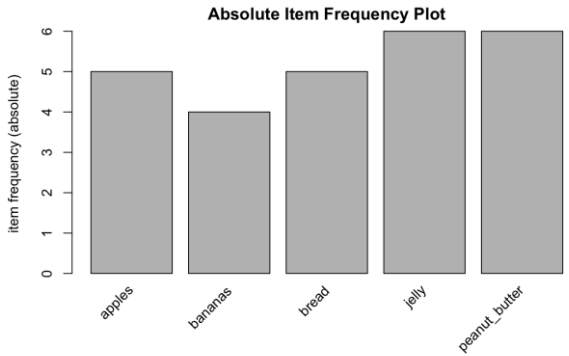
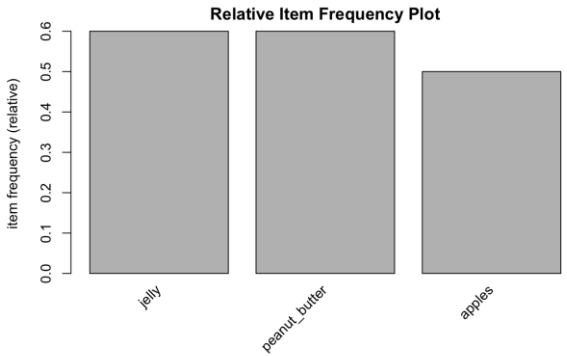
```r
tr <- read.transactions('mba.csv', header=F,format = 'basket', sep=',')
tr
summary(tr)
```




```
transactions as itemMatrix in sparse format with
 10 rows (elements/itemsets/transactions) and
 5 columns (items) and a density of 0.52

most frequent items:
        jelly peanut_butter        apples         bread       bananas       (Other)
            6             6             5             5             4             0

element (itemset/transaction) length distribution:
sizes
2 3
4 6

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    2.0     2.0     3.0     2.6     3.0     3.0

includes extended item information - examples:
```

MBA - R Implementation

- itemFrequency() calculates the frequency for each item in an itemMatrix
- itemFrequency() also used by itemFrequencyPlot() to produce a bar plot of item count frequencies or support

```r
```{r}
itemFrequency(tr)
itemFrequencyPlot(tr,type="absolute", main="Absolute Item Frequency Plot")
itemFrequencyPlot(tr,topN=3,type="relative",main="Relative Item Frequency Plot")
```
```

```
   apples   bananas     bread     jelly peanut_butter
      0.5       0.4       0.5       0.6           0.6
```





MBA - R Implementation

- apriori() finds all rules
- Takes parameters
  - maxlen = maximum size of mined frequent itemsets (default to 5)
- Output of the C implementation with timing
- Number of rules
- Most frequent items in the LHS and the RHS and length distributions
- Summary statistics of quality measures

```
```{r}
rules <- apriori(tr, parameter = list(supp=0.01, conf=0.8,maxlen=10))
summary(rules)
```
```

```
Apriori

Parameter specification:

Algorithmic control:

Absolute minimum support count: 0

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[5 item(s), 10 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 ... [10 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
set of 10 rules

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

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    2.0     2.0     2.5     2.5     3.0     3.0

summary of quality measures:
    support       confidence          lift           count
 Min.   :0.10   Min.   :0.8000   Min.   :1.333   Min.   :1.0
 1st Qu.:0.10   1st Qu.:0.8333   1st Qu.:1.458   1st Qu.:1.0
 Median :0.35   Median :1.0000   Median :1.833   Median :3.5
 Mean   :0.29   Mean   :0.9267   Mean   :1.794   Mean   :2.9
 3rd Qu.:0.40   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:4.0
 Max.   :0.50   Max.   :1.0000   Max.   :2.500   Max.   :5.0

mining info:
```

MBA - R Implementation

- inspect() allows to view the rules
- Can be customized for specific selection

```r
inspect(rules[1:10])
```

	lhs <fctr>		rhs <fctr>	support <dbl>	confidence <dbl>	lift <dbl>	count <int>
[1]	{bananas}	=>	{apples}	0.4	1.0000000	2.000000	4
[2]	{apples}	=>	{bananas}	0.4	0.8000000	2.000000	4
[3]	{bread}	=>	{peanut_butter}	0.4	0.8000000	1.333333	4
[4]	{peanut_butter}	=>	{jelly}	0.5	0.8333333	1.388889	5
[5]	{jelly}	=>	{peanut_butter}	0.5	0.8333333	1.388889	5
[6]	{bananas,bread}	=>	{apples}	0.1	1.0000000	2.000000	1
[7]	{apples,bread}	=>	{bananas}	0.1	1.0000000	2.500000	1
[8]	{bananas,jelly}	=>	{apples}	0.1	1.0000000	2.000000	1
[9]	{apples,peanut_butter}	=>	{jelly}	0.1	1.0000000	1.666667	1
[10]	{bread,jelly}	=>	{peanut_butter}	0.3	1.0000000	1.666667	3

1–10 of 10 rows

```r
inspect(head(rules,n=3,by='confidence'))
```

	lhs <fctr>		rhs <fctr>	support <dbl>	confidence <dbl>	lift <dbl>	count <int>
[1]	{bananas}	=>	{apples}	0.4	1	2.0	4
[2]	{bananas,bread}	=>	{apples}	0.1	1	2.0	1
[3]	{apples,bread}	=>	{bananas}	0.1	1	2.5	1

3 rows

MBA - R Implementation

- Specific product
- appearance

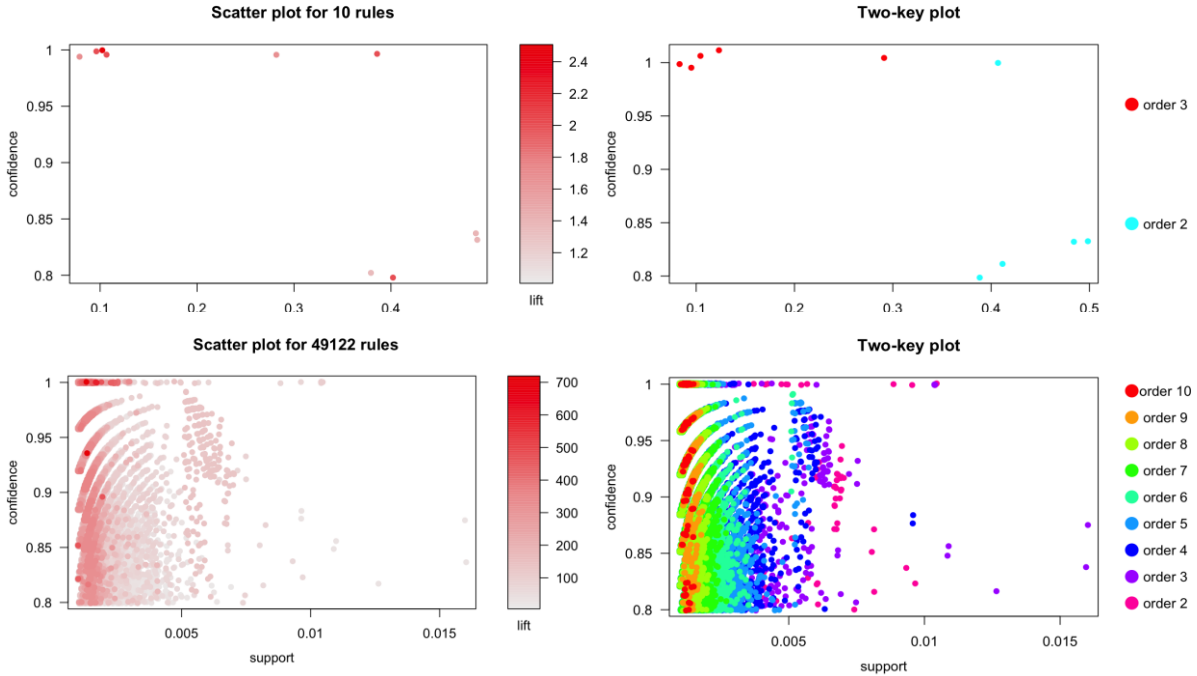
```r
bananas.rules <- apriori(tr, parameter = list(supp=0.01, conf=0.8),appearance =
list(default="lhs",rhs="bananas"))
bananas.rules <- apriori(tr, parameter = list(supp=0.01, conf=0.8),appearance =
list(lhs="bananas",default="rhs"))
```

```
Apriori

Parameter specification:

Algorithmic control:

Absolute minimum support count: 0

set item appearances ...[1 item(s)] done [0.00s].
set transactions ...[5 item(s), 10 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 ... [2 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
Apriori

Parameter specification:

Algorithmic control:

Absolute minimum support count: 0

set item appearances ...[1 item(s)] done [0.00s].
set transactions ...[5 item(s), 10 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 done [0.00s].
writing ... [1 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

MBA - R Implementation

- Scatter plot with two interest measures on the axes
- Rules with high lift have typically a relatively low support
- Special version - Two- key plot
  - “order” = the number of items contained in the rule

```r
plot(rules,jitter = 3)
plot(rules,jitter = 3,method="two-key plot")
```

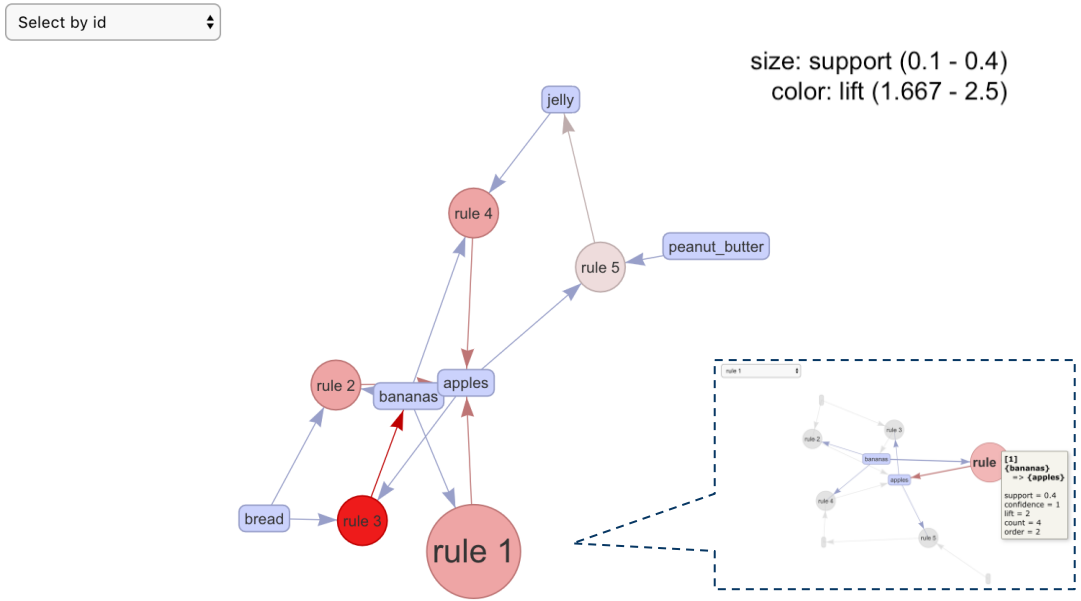


MBA - R Implementation

- Graph methods
  - vertices represent items
  - itemsets or rules as a second set of vertices
- Arrows pointing from items to rule vertices indicate LHS
- Arrow from a rule to an item indicates the RHS
- Interest measures added to the plot by using color/size of the itemsets/rules vertices

	lhs <fctr>	<fctr>	rhs <fctr>	support <dbl>	confidence <dbl>	lift <dbl>	count <int>
[1]	{bananas}	=>	{apples}	0.4	1	2.0	4
[2]	{bananas,bread}	=>	{apples}	0.1	1	2.0	1
[3]	{apples,bread}	=>	{bananas}	0.1	1	2.5	1

```{r}
top5rules <- head(rules, n = 5, by = "confidence")
plot(top5rules, method = "graph",  engine = "htmlwidget")
```

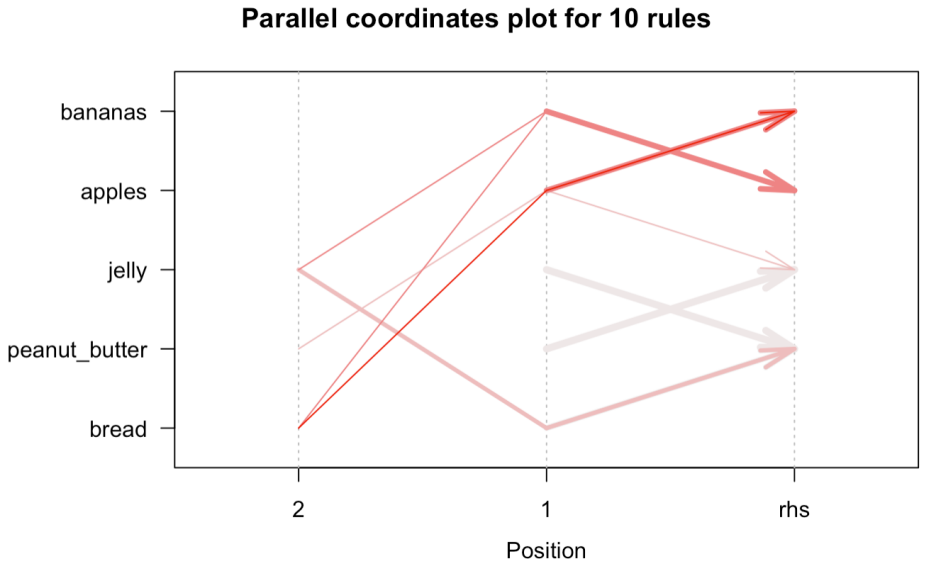


MBA - R Implementation

- Designed to visualize multidimensional data
- Each data point is represented by a line connecting the values for each dimension
- Items on the y-axis are nominal values
- x-axis represents the positions in a rule
- arrow is used where the head points to the consequent item

	lhs <fctr>		rhs <fctr>	support <dbl>	confidence <dbl>	lift <dbl>	count <int>
[1]	{bananas}	=>	{apples}	0.4	1	2.0	4
[2]	{bananas,bread}	=>	{apples}	0.1	1	2.0	1
[3]	{apples,bread}	=>	{bananas}	0.1	1	2.5	1

```
```{r}
# Filter top 10 rules with highest lift
rules2<-head(rules, n=10, by="lift")
plot(rules2, method="paracoord")
```
```



Conclusion

Pros

- Easy interpretation
- Easy implementation
- Scalable
- Multiple applications

Cons

- Support calculation is expensive
- Computationally expensive to find candidate rules
- Does not work with continuous variables

Resources

https://webfocusinfocenter.informationbuilders.com/wfappent/TLs/TL_rstat/source/marketbasket49.htm
https://bcssp10.files.wordpress.com/2013/02/lecture191.pdf
https://cran.r-project.org/web/packages/arules/arules.pdf
https://cran.r-project.org/web/packages/arules/vignettes/arules.pdf
https://cran.r-project.org/web/packages/arulesViz/vignettes/arulesViz.pdf
https://upcommons.upc.edu/bitstream/handle/2117/109798/129057.pdf?sequence=1&isAllowed=y
https://jmhldotorg.files.wordpress.com/2013/11/slidesinfosysbangaloreintroductionmba2011.pd
https://www.linkedin.com/pulse/introduction-market-basket-analysis-association-rule-abhishek-kumar/f

AR Algorithms

APRIORI : Apriori Algorithm proceeds by identifying the frequent individual items in the database and then extends them to larger and larger item sets as long as those item sets appear sufficiently often in the database

Eclat - uses a vertical database layout i.e. instead of explicitly listing all transactions ; each item is stored together with its cover (also called tidlist) and uses the intersection based approach to compute the support of an item set. It requires less space but it is suitable for small datasets

FP-growth : generates frequent item set without candidate generation. It uses a divide and conquer strategy while creating a tree based structure