An Introduction to Association Rules for Recommendation Systems

Agenda

- Association rules for recommender systems
- Association Rule Basics
- Measures of Utility
- Pitfalls of rule techniques
- Pros and Cons of Systems
- Some Systems & Libraries for Association rules

Association rules for recommender systems

- Recommender Systems are used by many companies e.g. Amazon, Netflix, Spotify

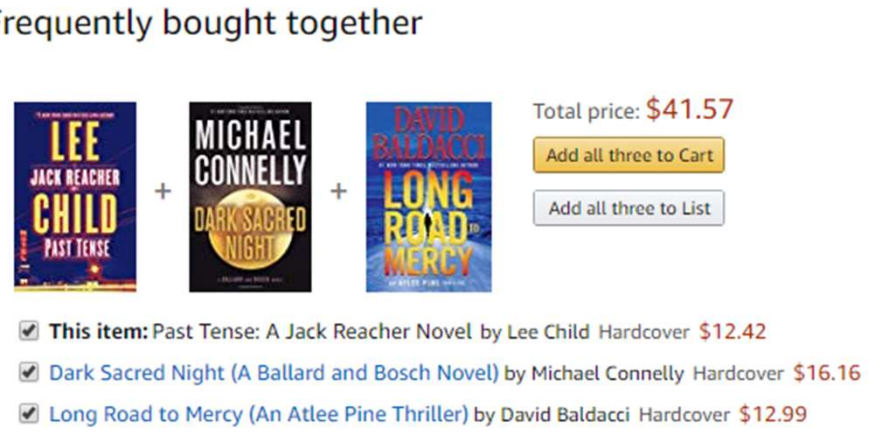
- How Association Rules differ:
  - Not regression nor typical classification method
  - Not collaborative and content-based filtering methods (see other presentation)
  - No user profile or item description needed
  - Not sequence mining, typically does not consider the order of transactions (can use weighting if desired)
- Application
  - Market basket analysis, website recommendations, intrusion detection, bioinformatics, etc

Association Rule Basics

- What is rule?
  - “one of a set of explicit or understood regulations or principles governing conduct within a particular activity or sphere”
  - A declaration of a scripted response that maximizes the likelihood of an outcome
- Association Rule
  - From itemset A-> itemset B
  - Database transaction
  - LHS -> RHS
  - People who do this, do that

Associative Rules Evaluation

- Associative rules effectiveness can be evaluated by these measures:
    1. Support
    2. Confidenc
    3. Lift



- Associative rules help lead the customer towards other similar products that they have the best chance of buying.

Example Dataset

	Beer	Bread	Milk	Diapei	Eggs	Coke
T_1	0	1	1	0	0	0
T_2	1	1	0	1	1	0
T_3	1	0	1	1	0	1
T_4	1	1	1	1	0	0
T_5	0	1	1	1	0	1

Support

$$Support = \frac{Number\ of\ transactions\ with\ both\ A\ and\ B}{Total\ number\ of\ transactions} = P(A \cap B)$$

- Support is an indication of how frequently the itemset appears in the dataset.
- A rule needs a **support** level (probability) before it can be considered statistically significant, and datasets often contain thousands or millions of transactions.

- Support
  - Indicates how frequently an item-set appears in the data set.

$$\text{supp}(X) = \frac{|\{t \in T; X \subseteq t\}|}{|T|}$$

  - Support of X with respect to T is defined as the ratio of transactions t in the dataset which contains the item-set X.

Support

- Support can be calculated as the fraction of rows containing both A and B or joint probability of A and B.
- Support means how much historical data supports your rule.
- The rule that buying diapers and beer implies buying milk has a support value of 2/5. You would count how many rows have all three divided by the total number of rows.

	Beer	Bread	Milk	Diapeı	Eggs	Coke
T_1	0	1	1	0	0	0
T_2	1	1	0	1	1	0
T_3	1	0	1	1	0	1
T_4	1	1	1	1	0	0
T_5	0	1	1	1	0	1

Confidence

$$Confidence = \frac{Number\ of\ transactions\ with\ both\ A\ and\ B}{Total\ number\ of\ transactions\ with\ A} = \frac{P(A \cap B)}{P(A)}$$

- Confidence is an indication of how often the rule has been found to be true.

- Confidence
  - Indicates how often a rule has been found to be true.

$$\text{conf}(X \Rightarrow Y) = \text{supp}(X \cup Y)/\text{supp}(X)$$

  - The confidence value of a rule, X -> Y, with respect to a set of transactions T, is the ratio of the transactions that contains X which also contains Y.

- Confidence is an estimate of the conditional probability $P(E_Y|E_X)$, the probability of finding the RHS of the rule in transactions under the condition that these transactions also contain the LHS.

Confidence

- Confidence is the fraction of rows containing B or conditional probability of B given A
- Confidence means how confident we are that the rule holds
- The confidence that if someone buys diapers and beer will buy milk is 2/3.

	Beer	Bread	Milk	Diapeı	Eggs	Coke
T_1	0	1	1	0	0	0
T_2	1	1	0	1	1	0
T_3	1	0	1	1	0	1
T_4	1	1	1	1	0	0
T_5	0	1	1	1	0	1

Lift

$$ExpectedConfidence = \frac{Number\ of\ transactions\ with\ B}{Total\ number\ of\ transactions} = P(B)$$

$$Lift = \frac{Confidence}{Expected\ Confidence} = \frac{P(A \cap B)}{P(A).P(B)}$$

- the ratio of the observed support to that expected if X and Y were [independent]
  - lift = 1, two events are independent of each other, no rule can be drawn involving those two events.
  - lift > 1, that lets us know the degree to which those two occurrences are dependent on one another, and makes those rules potentially useful for predicting the consequent in future data sets.
  - lift < 1, items are substitute to each other. This means that presence of one item has negative effect on presence of other item and vice versa.
- The value of lift is that it considers both the support of the rule and the overall data set.

- Lift
  - Indicates the ratio of observed support to that expected if X and Y were independent.

$$\text{lift}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X) \times \text{supp}(Y)}$$

Conviction – Lesser Used

- The ratio of the expected frequency that X occurs without Y (the frequency that the rule makes an incorrect prediction) if X and Y were independent divided by the observed frequency of incorrect predictions.

$$\mathrm{conv}(X \Rightarrow Y) = \frac{1 - \mathrm{supp}(Y)}{1 - \mathrm{conf}(X \Rightarrow Y)}$$

Pitfalls of rule techniques

- Confidence could be misleading:
- E.g. Dataset

1. Iphone, Headset
2. Iphone, Headset
3. Iphone
4. Iphone

- Conf(iPhone->Headset) = 2/4 = 0.5
- Conf(Headset->iPHone) = 2/2 = 1

- Headset->Iphone recommendation has higher confidence but it is not realistic.

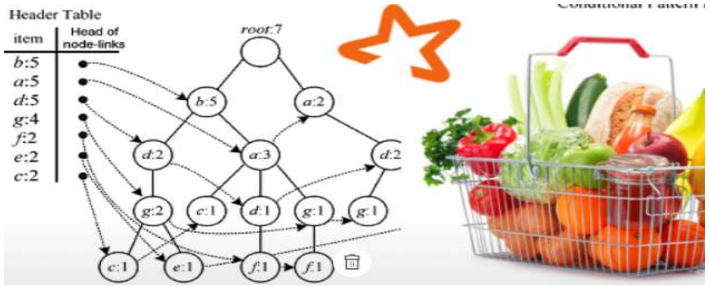
Pros and Cons

- Pros
  - Easy to understand & implement
  - Can be parallelized
  - Avoid cold-start problem with data analysis technique
  - Just transaction, probably don’t have privacy issue
- Cons
  - Computational expensive
  - May have fewer meaningful founding comparing to cross –domain recommender system

Some Association Libraries, Packages

- Other algorithms & libraries
  - Apriori (breath-first search, library “arules” in R)
  - Eclat, stands for equivalence class transformation (depth-first search, library “arules” in R), (eclat and FP=Growth)
  - FP-Growth (Apache Spark)
- “Recommenderlab” in R, bundle of algorithms

FP-Growth (Apache Spark)




```
# Load training data
df <- selectExpr(createDataFrame(data.frame(rawItems = c(
  "1,2,5", "1,2,3,5", "1,2"
))), "split(rawItems, ',') AS items")

fpm <- spark.fpGrowth(df, itemsCol="items", minSupport=0.5,
minConfidence=0.6)

# Extracting frequent itemsets
spark.freqItemsets(fpm)

# Extracting association rules
spark.associationRules(fpm)

# Predict uses association rules to and combines possible consequents
predict(fpm, df)
```

“Recommenderlab” in R

```
#evaluate multiple algorithms at once
algorithms <- list(
"association rules" = list(name = "AR",
param = list(supp = 0.01, conf = 0.01)),
"random items" = list(name = "RANDOM", param = NULL),
"popular items" = list(name = "POPULAR", param = NULL),
"item-based CF" = list(name = "IBCF", param = list(k = 5)),
"user-based CF" = list(name = "UBCF",
param = list(method = "Cosine", nn = 500))
)
results <- recommenderlab::evaluate(scheme,
algorithms,
type = "topNList",
n = c(1, 3, 5, 10, 15, 20)
)
```

https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

Consider

- http://mhahsler.github.io/arules/#:~:text=The%20arules%20package%20for%20R%20provides%20the%20infrastructure,and%20patterns%20using%20frequent%20itemsets%20and%20association%20rules.
- https://www.softwaretestinghelp.com/fp-growth-algorithm-data-mining/
- https://www.researchgate.net/publication/262325976_A_Bayesian_Association_Rule_Mining_Algorithm#:~:text=Two%20interesting-ness%20measures%20of%20association%20rules%3A%20Bayesian%20confidence,outputs%20best%20rules%20according%20to%20BC%20and%20BL.
- https://towardsdatascience.com/association-rule-mining-in-r-ddf2d044ae50#:~:text=Michael%20Hahsler%2C%20et%20al.%20has%20authored%20and%20maintains,following%20commands%20to%20install%20them.%20%3E%20install.packages%20%28%22arules%22%29
- Your web search, data science central, Open Source Courses