Recommender Systems
Part 1: Association Rules

Aram Balagyozyan

Department of Operations and Information Management
Kania School of Management
The University of Scranton

July 23, 2020

Introduction

- Recommender (or recommendation) systems are a critically important part of online vendors that offer a large variety of products or services. Examples include Amazon.com, which offers millions of different products; Netflix has thousands of movies for rental; Google searches over huge numbers of web pages; Internet radio websites such as Spotify and Pandora include a large variety of music albums by various artists; travel websites offer many destinations and hotels; social network websites have many groups.
- The recommender engine provides personalized recommendations to a user based on the user's information as well as on similar users' information. Information means behaviors indicative of preference, such as purchase, ratings, and clicking.
- The value that recommendation systems provide to users helps online companies convert browsers into buyers, increase cross-selling, and build loyalty.

Introduction

- This module is dedicated to two very popular recommender methods:
  1. **Association Rules** (a.k.a Affinity Analysis or Market Basket Analysis)
  2. **Collaborative Filtering**
- Part 1 of the lecture slides is dedicated to the former while Part 2 will be dedicated to the latter.
- **Association rule** discovery is aimed at discovering in transaction-type databases which groups of products that tend to be purchased together.
- In **collaborative filtering**, the goal is to provide personalized recommendations that leverage user-level information. This algorithm looks for users who have purchased a similar set of items or ranked items in similar fashion, and makes a recommendation to the initial user based on what the similar users purchased or liked.

Introduction to Association Rules

- Put simply, association rules constitute a study of "what goes with what."
- While being very popular in retail, association rules are also very useful in other fields such as medicine where we might want to learn what symptoms appear together.
- Association rules provide information in the form of "if-then" statements, such as for example "**if** a customer has purchased"Harry Potter,**"** **then** she is likely to also purchase "His Dark Materials."
- Note however, unlike if-then rules in logic, association rules are probabilistic in nature, since they are computed from the data.

Example: Artificial Data on Purchases of Phone Faceplates

- We start with an artificial example to demonstrate the concepts, computations, and steps of association rules. A more realistic case will be introduced afterwards.
- A store that sells accessories for cell phones runs a promotion on faceplates. Customers who purchase multiple faceplates from a choice of six different colors get a discount.
- The store manager would like to know what colors of faceplates customers are likely to purchase together.
- The next slides displays the database of all transactions.

Example: Artificial Data on Purchases of Phone Faceplates

Transaction	Faceplate Color Purchased
1	Red, White, Green
2	White, Orange
3	White, Blue
4	Red, White, Orange
5	Red, Blue
6	White, Blue
7	Red, Blue
8	Red, White, Blue, Green
9	Red, White, Blue
10	Yellow

Association Terminology

- The idea behind association rules is to examine all possible rules between items in an if-then format and select only those that are most likely to be indicators of true dependence. For example, the first transaction in the table above can be viewed as rule "IF *red* THEN *white and green*" or as another rule IF *red* THEN *white*.
- Term **antecedent** (or **A**, aka left-hand-side,**LFS**) describes the IF part and includes any number of unique items.
- Term **consequent** (or **C**, aka right-hand-side, **RHS**) describes the THEN part that may include any number of unique items not in the antecedent. Consequent may contain no items.
- Term **itemset** describes any possible combination of unique items purchased by a consumer including single items. Another term for itemset is **basket** of unique items. Each *antecedent* and *consequent* consist of an *itemset*.

Association Terminology

- For example, the first entry in the transaction table above, can be viewed as rule "if *red* then *white and green*" or as another rule "if *red* then *white*."
- In rule "if *red* then *white*", the antecedent is *red* and the consequent is *white*. The antecedent and consequent each contain a single item.
- In rule "if *red* then *white and green*," the antecedent is *red* and the consequent contains itemset {*white*,*green*}.

Generating Candidate Rules

- Ideally, we would like to consider all the rules that would be candidates for indicating association between items.
- However, all possible combinations include all possible associations between single items, pairs of items, triplets of items, and so on. Generating all these combinations may require a long computation time that grows exponentially with the number of items that consumers may purchase (p). Many of these co-occurrences may simply be due to chance rather than to a generalized pattern.
- A practical solution is to consider only combinations that occur with higher frequency in the dataset. Those are called *frequent itemsets*.
- In order to retain only the *frequent itemsets*, we may place a constraint that a given rule must apply to at least a certain percentage of the data. This is called the **support** of the association.

Generating Candidate Rules

- More formally, **support** of a rule is the percentage of transactions that include both the antecedent and consequent itemsets.
- Thus, if the rule is generally expressed as "if A then C," its support can be expressed as:

$$\mathit{Support} = P(\text{A AND C})$$

- In the above equation, the `P()` operator can be interpreted as both, the percentage (or proportion) of records contain both A and B and the probability of A and B occurring in transaction data.
- In our example, the support of the rule "if red then white" is equal to the percentage of records that contain both red and white phone faceplates: $100 \times \frac{4}{10} = 40\%$.
- If our chosen threshold was lower than 40%, then we would label the itemsets in the rule as *frequent itemsets*.

The Apriori Algorithm of Generating Frequent Itemsets

- One of the classic approaches to generating frequent itemsets is the recursive *Apriory Algorithm* that is relatively fast.
- **Step 1** For each one-itemset, the algorithm obtains the percentage of transactions in the database that include the item (that is the support). The transactions that have support above the desired minimum support are kept and the remaining one-itemsets are dropped.
- **Step 2** Using the most frequent one-itemset from Step 1, the algorithm generates all two-itemsets and for each two-itemset again obtains the percentage of transactions in the database that include the two-itemset (support). If the support is greater than the desired minimum support, the two-itemset is kept, otherwise it is dropped.
- **Steps 3, 4, . . . k** The above procedure is repeated to obtain the most frequent three-itemsets, then four-itemsets, and so on. In general, generating *k*-itemsets uses the most frequent (*k-1*)-itemsets that were retained in the preceding step.

Measuring the Strength of a Rule: Confidence

- The fact that two items are purchased together is not necessarily indicative of the strength of a rule.
- For example, whenever I do my groceries, I buy both, milk and tomatoes. This co-occurrence of milk and tomatoes may be flagged by the apriori algorithm as a frequent itemset. However, significant support of the itemset {milk, tomatoes} should not necessarily be suggestive of a rule *"if milk then tomatoes"* (or *"if tomatoes then milk"*).

Measuring the Strength of a Rule: Confidence

- **Confidence** is a measure the certainty about the if-then rule.
- In this context, confidence not to be confused with statistical confidence. It is defined as a ratio of the ratio of transactions that include all antecedent and consequent itemsets (namely the support) to the ratio of transactions that include all the antecedent itemsets.

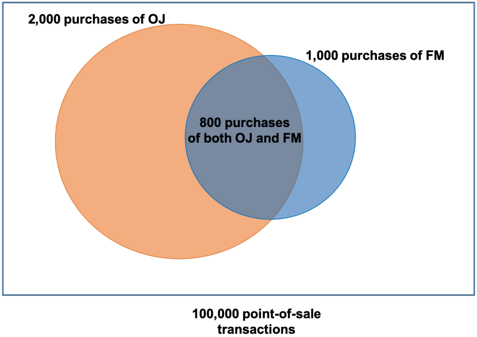
$$\mathit{Confidence} = \frac{\text{P(A AND C)}}{\text{P(A)}}$$

- Notice that confidence of a rule can be interpreted as the conditional probability of observing the consequent given that we observed the antecedent.

$$\mathit{Confidence} = \frac{\text{P(A AND C)}}{\text{P(A)}} = P(C|A)$$

Measuring the Strength of a Rule: Confidence

- To understand the intuition behind the confidence measure consider an example of a supermarket database with 100,000 point-of-sale transactions.
- Of these transactions, 2,000 include orange juice (OJ), 1,000 include flu medication (FM). In addition, of the 2,000 purchases of OJ and 1,000 purchases of FM, 800 purchases include both OJ and FM.



Measuring the Strength of a Rule: Confidence

- *Support* for orange juice and for flu medicine are 2,000/100,000 = 2% and 1,000/100,000 = 1% respectively.
- *Confidence* of the “if orange juice then flu medication” rule is 800/2,000 = 40% and of the “if flu medication then orange juice” rule is 800/1,000 = 80%.
- The higher is the value of confidence the stronger is the association rule.
- One big problem with the measure of confidence is that if the antecedent and /or consequent has a high level of support, we may end up having high level of confidence even when the antecedent and consequent are independent.
- Consider an example of a supermarket where all consumers always buy bananas. Thus, all 100,000 transactions contain bananas and of those, 1,000 contain purchases of flu medicine.
- Confidence of the “if flu medicine then bananas” rule is 1,000/1,000 = 100%. However, such high confidence is “*not surprising*” and should not be taken as strength of a rule.

Measuring the Strength of a Rule: Lift Ratio

- We would like the association to be in some sense "surprising." There are many notions of surprisingness and the *lift ratio* is one of the most popular and intuitive ones.
- The lift ratio measures how much more frequently does an association occur than we would expect by chance.
- The lift ratio is calculated simply as

$$Lift(A, C) = \frac{P(\text{A AND C})}{P(\text{A}) \times P(\text{C})}$$

Measuring the Strength of a Rule: Lift Ratio

- In English, the lift ratio of the co-occurrence of the antecedent itemset and consequent itemset is the probability that we actually see the two together compared to the probability that we would see the two together if they were unrelated to (independent of) each other.
- A lift ratio greater than one is the factor by which seeing the antecedent "boosts" the likelihood of seeing the consequent as well.
- A lift ration less than one is indicative that the antecedent and consequent appear less often together than expected, this means that the occurrence of the antecedent has a negative effect on the occurrence of the consequent.

Lift Ratio Examples

- Using the numbers in the example above, the lift ratio of the co-occurrence of itemsets in the rule "if flu medication (FM) then bananas (B)" is:

$$Lift(FM, B) = \frac{P(\text{FM AND B})}{P(\text{FM}) \times P(\text{B})} = \frac{0.01}{0.01 \times 1} = 1$$

- The factor by which FM boosts the likelihood of seeing B is 0.
- For the numbers in the example before, the lift ratio of the "if orange juice then flu medication" rule is:

$$Lift(OJ, FM) = \frac{P(\text{OJ AND FM})}{P(\text{OJ}) \times P(\text{FM})} = \frac{0.008}{0.02 \times 0.01} = 40$$

- The likelihood of a purchase that includes both FM and OJ is 40 times than what it would have been if the two products were unrelated to each other. Put differently, the factor by which OJ boosts the likelihood of FM is 40.

Example: Purchases of Phone Faceplates

- Transaction data are displayed in one of the two formats: a transaction database (with each row representing the list of items purchased in a single transaction), or a binary incidence matrix.
- Suppose we want to obtain association rules between items that have a support percentage of at least 20% (or 2 transactions in this case).

Example: Purchases of Phone Faceplates

- First prepare the dataset.

```
# load the arules library that will do the job
library(arules)
# load the data
fp.df <- read.csv("Faceplate.csv")
# remove the first column and convert to matrix
fp.mat <- as.matrix(fp.df[,-1])
fp.mat
# convert the binary incidence matrix into a
# transaction database
fp.trans <- as(fp.mat,"transactions")
inspect(fp.trans)
```

Example: Purchases of Phone Faceplates

- Below is the binary incidence matrix.

```
##       Red White Blue Orange Green Yellow
##  [1,]   1     1    0      0     1      0
##  [2,]   0     1    0      1     0      0
##  [3,]   0     1    1      0     0      0
##  [4,]   1     1    0      1     0      0
##  [5,]   1     0    1      0     0      0
##  [6,]   0     1    1      0     0      0
##  [7,]   1     0    1      0     0      0
##  [8,]   1     1    1      0     1      0
##  [9,]   1     1    1      0     0      0
## [10,]   0     0    0      0     0      1
```

Example: Purchases of Phone Faceplates

- Below is the binary incidence matrix converted into a transaction database.

```
##      items
## [1]  {Red,White,Green}
## [2]  {White,Orange}
## [3]  {White,Blue}
## [4]  {Red,White,Orange}
## [5]  {Red,Blue}
## [6]  {White,Blue}
## [7]  {Red,Blue}
## [8]  {Red,White,Blue,Green}
## [9]  {Red,White,Blue}
## [10] {Yellow}
```

Example: Purchases of Phone Faceplates

- While the `apriori()` function of the **arules** package is capable of selecting at once the best association rules, for demonstration purposes we will take a step-by-step approach.
- As a first step, we are interested in retaining only the frequent itemsets with percentage support exceeding a certain threshold. For the sake of this example, let the threshold be 20%.

```
# Run the apriori algorithm in the arules package
freq_is <- apriori(fp.trans,
                   parameter = list(
                   # look for frequent itemsets
                   target = "frequent itemsets",
                   support = 0.2))
# keep only the columns with itemsets, counts,
# and support
freq_is <- inspect(freq_is)[,c(1,4,2)]
# sort the list of frequent itemsets by the count
freq_is[order(freq_is$count, decreasing = TRUE),]
```

Example: Purchases of Phone Faceplates

- Below is the output of the last R statement above.

```
##                      items count support
## [5]                {White}     7     0.7
## [3]                 {Blue}     6     0.6
## [4]                  {Red}     6     0.6
## [9]             {Red,Blue}     4     0.4
## [10]          {White,Blue}     4     0.4
## [11]           {Red,White}     4     0.4
## [1]               {Orange}     2     0.2
## [2]                {Green}     2     0.2
## [6]         {White,Orange}     2     0.2
## [7]            {Red,Green}     2     0.2
## [8]          {White,Green}     2     0.2
## [12] {Red,White,Green}     2     0.2
## [13]  {Red,White,Blue}     2     0.2
```

- Itemset `{White}` has a support of 0.7, because 7 of the 10 transactions included a white faceplate. The last itemset `{Red, White, Blue}` has a support of 0.2, because only 2 of the 10 transactions included red, white, ans blue faceplates.

Example: Purchases of Phone Faceplates

- As the second step, we are interested in which association rules within frequent itemsets dominate.
- For the sake of discussion, let's concentrate on the 12^{th} itemset in the frequent itemset table on the previous slide `{RWG}` (in the interest of space, I will use the first letters of the colors and ignore commas).
- Since any subset (e.g. `{R}`) must occur at least as frequently as the set is belongs to (e.g. `{RW}`), each subset will be in the list of frequent itemsets also.
- It is then straight-forward to compute the confidence and support for each possible rule.
- Since the `apriori()` routine of the **arules** package generates only the top single-consequent rules, in what follows, I will also concentrate on the single-consequent rules.
- The following slide lists all the possible single-consequent rules of the `{RWG}` itemset and the corresponding confidence and lift ratios.

Example: Purchases of Phone Faceplates

Rule	Confidence	Lift
1. $RW \Rightarrow G$	$0.2/0.4 = 50\%$	$0.2/(0.4 \times 0.2) = 2.5$
2. $RG \Rightarrow W$	$0.2/0.2 = 100\%$	$0.2/(0.2 \times 0.7) = 1.43$
3. $WG \Rightarrow R$	$0.2/0.2 = 100\%$	$0.2/(0.2 \times 0.6) = 1.67$
4. $R \Rightarrow G$	$0.2/0.6 = 33.33\%$	$0.2/(0.6 \times 0.2) = 1.67$
5. $W \Rightarrow G$	$0.2/0.7 = 28.57\%$	$0.2/(0.7 \times 0.2) = 1.43$
6. $R \Rightarrow W$	$0.4/0.6 = 66.67\%$	$0.4/(0.6 \times 0.7) = 0.95$
7. $G \Rightarrow W$	$0.2/0.2 = 100\%$	$0.2/(0.2 \times 0.7) = 1.43$
8. $W \Rightarrow R$	$0.4/0.7 = 57.14\%$	$0.4/(0.7 \times 0.6) = 0.95$
10. $G \Rightarrow R$	$0.2/0.2 = 100\%$	$0.2/(0.2 \times 0.6) = 1.67$

Example: Purchases of Phone Faceplates

- While the table on the previous slide was build “by hand,” it would be quite tedious to go through all possible combinations of different itemsets and rules and calculate their confidence and support.
- The `apriori()` function of the **arules** package will do the job for you.

```
#re-run apriori() with more parameters
freq_is <- apriori(fp.trans,
                   parameter = list(
#no need to set 'target="rules", it is the default
                   support = 0.2, #min support
                   conf = 0.5) #min confidence
                 )
# inspect the first 8 rules sorted by their lift
inspect(head(sort(freq_is, by="lift"),n=8))
```

Example: Purchases of Phone Faceplates

- Below is a partial output of the code on the previous slide:

	lhs		rhs	support	confidence	lift
[1]	{Red,White}	=>	{Green}	0.2	0.50	2.50
[2]	{Green}	=>	{Red}	0.2	1.00	1.67
[3]	{White,Green}	=>	{Red}	0.2	1.00	1.67
[4]	{Orange}	=>	{White}	0.2	1.00	1.43
[5]	{Green}	=>	{White}	0.2	1.00	1.43
[6]	{Red,Green}	=>	{White}	0.2	1.00	1.43
[7]	{Blue}	=>	{Red}	0.4	0.67	1.11
[8]	{Red}	=>	{Blue}	0.4	0.67	1.11

Interpreting the Results of the `apriori()` Function

- We can translate each of the rules from the table above into an understandable sentence that provides information about rule performance.
- For example, rule number 4 states that if orange is purchased then with confidence of 100% white will also be purchased. In addition, the factor by which orange boosts the likelihood of white is 1.43. That is, since `P(W)=0.7` and `P(W|O)=1`, if orange is purchased, the likelihood that white will be purchased increases from 0.7 to 1, or by a factor of 1.43.

Interpreting the Results of the `apriori()` Function

- A lift ratio indicates how efficient the rule is in finding cosequents, compared to random selection.
- A very efficient rule is preferred to an inefficient rule, but we must still consider support: A very efficient rule that has very low support may not be as desirable as a less efficient rule with much greater support.
- We must also consider confidence. The confidence tells us at what rate consequents will be found given antecedents. It is useful in determining the business or operational usefulness of a rule: A rule with low confidence may find consequents at too low a rate to be worth the cost of (say) promoting the consequent in all the transactions that involve the antecedent.
- Finally, be aware of the possibility that a rule may be found to be strong purely because of chance rather than some true underlying association.

Interpreting the Results of the `apriori()` Function

- Two principles can guide us in assessing rules for possible randomness due to chance effects.
  1. The more records a single rule is based on, the more solid the conclusion.
  2. The greater the number of distinct rules we consider, the more likely it is that at least some will be based on chance sampling results. If we consider a very large number of distinct rules based on a relatively small-sized transaction database, it will be quite likely that at least some of the rules are generated by pure chance.
- A reasonable approach to the second issue above is to consider rules from the top-down in terms of business or operational applicability, and not consider more than what can reasonably be incorporated in a human decision-making process.

Association Rules: A Real-life Example

- The `Groceries` data set that comes with the **arules** package contains 1 month (30 days) of real-world point-of-sale transaction data from a typical local grocery outlet. The data set contains 9835 transactions and the items are aggregated to 169 categories.
- Hypothetically, the manager may want to establish a data-driven promotion policy. To that end, he would like to know what strong association rules exist between the grocery itemsets.

Association Rules: A Real-life Example

- We will first load the Grocery transaction data, take a quick look at it, and using the `apriori()` routine generate rules.

```r
data("Groceries") # load the data
inspect(head(Groceries, 4))# display top 3 trans.
```

```
##     items
## [1] {citrus fruit,
##      semi-finished bread,
##      margarine,
##      ready soups}
## [2] {tropical fruit,
##      yogurt,
##      coffee}
## [3] {whole milk}
## [4] {pip fruit,
##      yogurt,
##      cream cheese ,
##      meat spreads}
```

Association Rules: A Real-life Example

- To plot the frequency of all the items, we could use the `itemFrequencyPlot(Groceries)` command, which is a utility of the **arules** package. However, with 169 categories in the dataset, the horizontal axis becomes overwhelmed with items.
- Instead, we could piece by piece build the plot of the relative frequency of the top say 20 items in the Grocery database using the `itemFrequency()` function of the **arules** package and `ggplot()` of the **ggplot2** package.

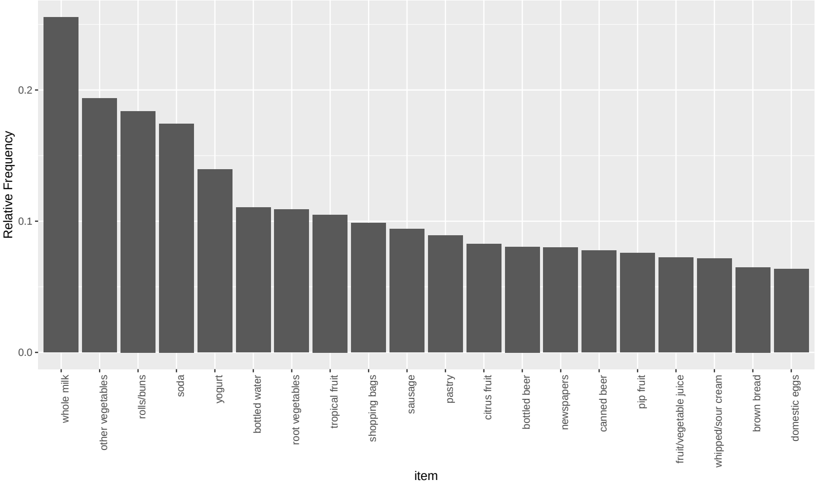
Association Rules: A Real-life Example

See below.

```
library(dplyr) # load dplyr for data manipulation
library(ggplot2) # load ggplot2
iF20 = tail(sort(itemFrequency(Groceries)), 20)
# sort(itemFrequency(Groceries)) sorts items by
# frequency;
# tail(..,20) keeps only the bottom (most frequent) 20
# We need to convert the iF20 frequency table into
# a data frame
if.df <- data.frame(freq=iF20)
# then add a column with item names to the data frame.
if.df <- mutate(if.df, item=row.names(if.df))
# then plot
ggplot(if.df , aes(x = reorder(item,-freq),y = freq))+
# reorder(item,-freq) orders item by frequency
geom_col()+ ylab("Relative Frequency") + xlab("item")+
theme(axis.text.x = element_text(angle = 90, hjust=1))
```

Association Rules: A Real-life Example

Below is the resulting chart that shows that whole milk (with P(Milk) $= 0.256$) was the most frequently purchased item in the `Grocery` dataset.



Association Rules: A Real-life Example

- Finally, we can filter out the strongest rules by running the `apriori()` routine

```
rules <- apriori (Groceries,
                  parameter = list(
                    supp = 0.01,
                    conf = 0.5))
# 'high-lift' rules.
rules <- sort(rules, by= "lift", decreasing= TRUE)
# display the rules
inspect(head(rules,7))
```

Association Rules: A Real-life Example

- Below is partial output of the code above:

	lhs		rhs	support	confidence	lift
[1]	{citrus fruit,root vegetables}	=>	{other vegetables}	0.01	0.59	3.03
[2]	{tropical fruit,root vegetables}	=>	{other vegetables}	0.01	0.58	3.02
[3]	{root vegetables,rolls/buns}	=>	{other vegetables}	0.01	0.50	2.59
[4]	{root vegetables,yogurt}	=>	{other vegetables}	0.01	0.50	2.58
[5]	{curd,yogurt}	=>	{whole milk}	0.01	0.58	2.28
[6]	{other vegetables,butter}	=>	{whole milk}	0.01	0.57	2.24
[7]	{tropical fruit,root vegetables}	=>	{whole milk}	0.01	0.57	2.23

- One may argue that rules 1-4 are of little use; if one purchases certain types of vegetables, there is obviously a high probability that they will also purchase "other vegetables."
- Rules 5-7, however, may be interesting. For example, Rule 5 states that if a customer buys curd and yogurt, the probability that she will also pick up whole milk is 0.57 which is 2.27 times higher than the unconditional probability that a customer buys whole milk by itself (recall that P(whole milk) = 0.256).