CHAPTER 6

Association Analysis

The beer and diaper association story in the analytics circle is (urban) legendary (Power, 2002). There are many variations of this story, but the basic plot is that a supermarket company discovered that customers who buy diapers also tend to buy beer. The beer and diaper relationship heralded unusual, unknown, and quirky nuggets that could be learned from the transaction data of a supermarket. How did the supermarket determine that such a relationship between products existed? The answer: Data Science (It was called data mining back in the 2000s). Specifically, association analysis.

Association analysis measures the strength of co-occurrence between one item and another. The objective of this class of data science algorithms is not to predict an occurrence of an item, like classification or regression algorithms do, but to find usable patterns in the co-occurrences of the items. Association rules learning is a branch of unsupervised learning processes that discover hidden patterns in data, in the form of easily recognizable *rules*.

Association algorithms are widely used in retail analysis of transactions, recommendation engines, and online clickstream analysis across web pages, etc. One of the popular applications of this technique is called *market basket analysis*, which finds co-occurrences of one retail item with another item within the same retail purchase transaction (Agrawal, Imieliński, & Swami, 1993). If patterns within transaction data tell us that baby formula and diapers are usually purchased together in the same transaction, a retailer can take advantage of this association for bundle pricing, product placement, and even shelf space optimization within the store layout. Similarly, in an online business setting, this information can be leveraged for real-time cross selling, recommendations, cart offers, and post-purchase marketing strategies. The results of association analysis are commonly known, for example a burger with fries or baby formula with diapers; however, uncommon relationships are the

Data Science. DOI: https://doi.org/10.1016/B978-0-12-814761-0.00006-X

prized discoveries, the ones businesses can take advantage of. The downside is that association analysis may also yield spurious relationships between items. When dealing with data containing billions of transactions, transactions with all kinds of possibilities with strange combinations of itemsets (e.g., nicotine patch and cigarettes) can be found. It takes analytical skill and business knowledge to successfully apply the outcomes of an association analysis. The model outcome of an association analysis can be represented as a set of rules, like the one below:

{Item A} → {Item B}

This rule indicates that based on the history of all the transactions, when Item A is found in a transaction or a basket, there is a strong propensity of the occurrence of Item B within the *same* transaction. Item A is the *antecedent* or *premise* of the rule and Item B is the *consequent* or *conclusion* of the rule. The antecedent and consequent of the rule can contain more than one item, like Item A and Item C. To mine these kinds of rules from the data, previous customer purchase transactions would need to be analyzed. In a retail business, there would be millions of transactions made in a day with thousands of stock keeping units, which are unique for an item. Hence, two of the key

CROSS SELLING: CUSTOMERS WHO BOUGHT THIS ALSO BOUGHT. . .

Consider an e-commerce website that sells a large selection of products online. One of the objectives of managing an e-commerce business is to increase the average order value of a visit. Optimizing order size is even more critical when the businesses pay for acquisition traffic through search engine marketing, online advertisements, and affiliate marketing. Businesses attempt to increase average order value by cross-selling and up-selling relevant products to the customer, based on what they have purchased or are currently purchasing in the current transaction (a common fast-food equivalent: "Do you want fries with the burger?"). Businesses need to be careful by weighing the benefit of suggesting an extremely relevant product against the risk of irritating a customer who is already making a transaction. In a business where there are limited products (e.g., fast-food industry), cross-selling a product with another product is straightforward and is quite embedded in the business. But, when the number of unique products runs into the thousands or millions, determining a set of *affinity products* when customers are looking at a product is quite challenging.

To better learn about product affinity, understanding purchase history data is helpful. The information on how one product creates affinity to another product relies on the fact that both the products appear in the same transaction. If two products are bought together, then it can be hypothesized, that the necessity of those products arise simultaneously for the customer. If the two products are bought together many times, by a large number of customers, then there is a strong possibility of an affinity pattern within these products. In a new later transaction, if a customer picks one of those affinity products, then there is an increased likelihood that the other product will be picked by the customer, in the same transaction.

The key input for affinity analysis is a list of past transactions with product information. Based on the analysis of these transactions, the most frequent product pairs can

(*Continued*)

(Continued)

be determined. A threshold will need to be defined for "frequent" because a few appearances of a product pair doesn't qualify as a pattern. The result of an affinity analysis is a rule set that says, "If product A is purchased, there is an increased likelihood that product B will be purchased in the same transaction." This rule set can be leveraged to provide cross sell recommendations on the product page of product A. Affinity analysis is the concept behind the web widgets which state, "Customers who bought this also bought..."

considerations of association analysis are computational time and resources. However, over the last two decades newer and more efficient algorithms have been developed to mitigate this problem.

6.1 MINING ASSOCIATION RULES

Basic association analysis just deals with the *occurrence* of one item with another. More advanced analysis can take into consideration the quantity of occurrence, price, and sequence of occurrence, etc. The method for finding association rules through data science involves sequential steps:

Step 1: Prepare the data in transaction format. An association algorithm needs input data to be formatted in transaction format $t_x = \{i_1, i_2, i_3\}$.
Step 2: Short-list frequently occurring *itemsets*. Itemsets are combinations of items. An association algorithm limits the analysis to the most frequently occurring items, so that the final rule set extracted in the next step is more meaningful.
Step 3: Generate *relevant* association rules from itemsets. Finally, the algorithm generates and filters the rules based on the interest measure.

To start with, consider a media website, like BBC or Yahoo News, with categories such as news, politics, finance, entertainment, sports, and arts. A session or transaction in this example is one visit for the website, where the same user accesses content from different categories, within a certain session period. A new session usually starts after 30 minutes of inactivity. Sessions are quite similar to transactions in a traditional brick and mortar model and the pages accessed can be related to items purchased. In online news sites, items are *visits* to the categories such as News, Finance, Entertainment, Sports, and Arts. The data can be collected as shown in Table 6.1, with a list of sessions and media categories accessed during a given session. The objective in this data science task is to find associations between media categories.

For association analysis of these media categories, a dataset in a particular transaction format would be needed. To get started with association analysis, it would be helpful to pivot the data in the format shown in Table 6.2.

Table 6.1 Clickstream

Session ID	List of Media Categories Accessed
1	{News, Finance}
2	{News, Finance}
3	{Sports, Finance, News}
4	{Arts}
5	{Sports, News, Finance}
6	{News, Arts, Entertainment}

Table 6.2 Clickstream Dataset

Session ID	News	Finance	Entertainment	Sports	Arts
1	1	1	0	0	0
2	1	1	0	0	0
3	1	1	0	1	0
4	0	0	0	0	1
5	1	1	0	1	0
6	1	0	1	0	1

This binary format indicates the presence or absence of article categories and ignores qualities such as minutes spent viewing or the sequence of access, which can be important in certain sequence analysis. For now, the focus is on basic association analysis and the terminologies used in association rules will be reviewed.

6.1.1 Itemsets

In the examples of association rules discussed so far, the antecedent and consequent of the rules had only one item. But, as mentioned before, they can involve multiple items. For example, a rule can be of the following sort:

$$\{\text{News}, \text{Finance}\} \rightarrow \{\text{Sports}\}$$

This rule implies that, if users have accessed news and finance in the same session, there is a high likelihood that they would also access sports articles, based on historical transactions. The combination of news and finance items is called an *itemset*. An itemset can occur either in the antecedent or in the consequent portion of the rule; however, both sets should be disjointed, which means there should not be any common item on both sides of the rule. Obviously, there is no practical relevance for rules like "News and Finance users are most likely to visit the News and Sports pages." Instead,

rules like "If users visited the Finance page they are more likely to visit the News and Sports pages" make more sense. The introduction of the itemset with more than one item greatly increases the permutations of the rules to be considered and tested for the strength of relationships.

The strength of an association rule is commonly quantified by the *support* and *confidence* measures of a rule. There are a few more quantifications like *lift* and *conviction* measures that can be used in special cases. All these measures are based on the relative frequency of occurrences of a particular itemset in the transactions dataset used for training. Hence, it is important that the training set used for rule generation is unbiased and truly represents a universe of transactions. Each of these frequency metrics will be elaborated on in the coming sections.

Support

The *support of an item* is simply the relative frequency of occurrence of an itemset in the transaction set. In the dataset shown in Table 6.2, support of {News} is five out of six transactions, $5/6 = 0.83$. Similarly, support of an itemset {News, Finance} is the co-occurrence of both news and finance in a transaction with respect to all the transactions:

$$\text{Support}(\{\text{News}\}) = 5/6 = 0.83$$
$$\text{Support}(\{\text{News, Finance}\}) = 4/6 = 0.67$$
$$\text{Support}(\{\text{Sports}\}) = 2/6 = 0.33$$

The *support of a rule* is a measure of how all the items in a rule are represented in the overall transactions. For example, in the rule $\{\text{News}\} \rightarrow \{\text{Sports}\}$, News and Sports occur in two of six transactions and hence, support for the rule $\{\text{News}\} \rightarrow \{\text{Sports}\}$ is 0.33. The support measure for a rule indicates whether a rule is worth considering. Since the support measure favors the items where there is high occurrence, it uncovers the patterns that are worth taking advantage of and investigating. This is particularly interesting for businesses because leveraging patterns in high volume items leads to incremental revenue. Rules with low support have either infrequently occurring items or an item relationship occurs just by chance, which may yield spurious rules. In association analysis, a threshold of support is specified to filter out infrequent rules. Any rule that exceeds the support threshold is then considered for further analysis.

Confidence

The *confidence of a rule* measures the likelihood of the occurrence of the consequent of the rule out of all the transactions that contain the antecedent of the rule. Confidence provides the reliability measure of the rule. Confidence of the rule $(X \rightarrow Y)$ is calculated by

$$\text{Confidence}\,(X \rightarrow Y) = \frac{\text{Support}\,(X \cup Y)}{\text{Support}\,(X)} \tag{6.1}$$

In the case of the rule {News, Finance} → {Sports}, the question that the confidence measure answers is, if a transaction has both News and Finance, what is the likelihood of seeing Sports in it?

$$\begin{aligned}\text{Confidence}\,(\{\text{News, Finance}\} \rightarrow \{\text{Sports}\}) &= \frac{\text{Support}\,(\{\text{News, Finance, Sports}\})}{\text{Support}\,(\{\text{News, Finance}\})} \\ &= \frac{2/6}{4/6} \\ &= 0.5\end{aligned}$$

Half of the transactions that contain News and Finance also contain Sports. This means that 50% of the users who visit the news and finance pages also visit the sports pages.

Lift

Though confidence of the rule is widely used, the frequency of occurrence of a rule consequent (conclusion) is largely ignored. In some transaction itemsets, this can provide spurious scrupulous rule sets because of the presence of infrequent items in the rule consequent. To solve this, the support of a consequent can be put in the denominator of a confidence calculation. This measure is called the *lift of the rule*. The lift of the rule can be calculated by

$$\text{Life}\,(X \rightarrow Y) = \frac{\text{Support}\,(X \cup Y)}{\text{Support}\,(X) \times \text{Support}\,(Y)}$$

In the case of our example:

$$\begin{aligned}\text{Life}\,\left(\{\text{News, Finance}\} \rightarrow \{\text{Sports}\}\right) &= \frac{\text{Support}\,(X \cup Y)}{\text{Support}\,(X) \times \text{Support}\,(Y)} \\ &= \frac{0.333}{0.667 \times 0.33} = 1.5\end{aligned} \tag{6.2}$$

Lift is the ratio of the observed support of {News and Finance} and {Sports} with what is expected if {News and Finance} and {Sports} usage were completely independent. Lift values closer to 1 mean the antecedent and consequent of the rules are independent and the rule is not interesting. The higher the value of lift, the more interesting the rules are.

Conviction

The *conviction of the rule* $X \rightarrow Y$ is the ratio of the expected frequency of X occurring in spite of Y and the observed frequency of incorrect predictions. Conviction takes into account the direction of the rule. The conviction of $(X \rightarrow Y)$ is not the same as the conviction of $(Y \rightarrow X)$. The conviction of a rule $(X \rightarrow Y)$ can be calculated by

$$\text{Conviction}\,(X \rightarrow Y) = \frac{1 - \text{Support}\,(Y)}{1 - \text{Confidence}\,(X \rightarrow Y)}$$

For our example, (6.3)

$$\text{Conviction}\big(\{\text{News}, \text{Finance}\} \rightarrow \{\text{Sports}\}\big) = \frac{1 - 0.33}{1 - 0.5} = 1.32$$

A conviction of 1.32 means that the rule ({News, Finance} → {Sports}) would be incorrect 32% more often if the relationship between {News, Finance} and {Sports} is purely random.

6.1.2 Rule Generation

The process of generating meaningful association rules from the dataset can be broken down into two basic tasks.

1. *Finding all frequent itemsets.* For an association analysis of n items it is possible to find $2^n - 1$ itemsets excluding the null itemset. As the number of items increase, there is an exponential increase in the number of itemsets. Hence it is critical to set a minimal support threshold to discard less frequently occurring itemsets in the transaction universe. All possible itemsets can be expressed in a visual lattice form like the diagram shown in Fig. 6.1. In this figure one item {Arts} is excluded from the itemset generation. It is not uncommon to exclude items so that the association analysis can be focused on subsets of important relevant items. In the supermarket example, some filler items like grocery bags can be excluded from the analysis. An itemset tree (or lattice) helps demonstrate the methods to easily find frequent itemsets.
2. *Extracting rules from frequent itemsets.* For the dataset with n items it is possible to find $3^n - 2^{n+1} + 1$ rules (Tan, Steinbach, & Kumar, 2005). This step extracts all the rules with a confidence higher than a minimum confidence threshold.

This two-step process generates hundreds of rules even for a small dataset with dozens of items. Hence, it is important to set a reasonable support and confidence threshold to filter out less frequent and less relevant rules in the search space. The generated rules can also be evaluated with support, confidence, lift, and conviction measures. In terms of computational requirements, finding all the frequent itemsets above a support threshold is more expensive than extracting the rules. Fortunately, there are some algorithmic approaches to efficiently find the frequent itemsets. The Apriori and Frequent Pattern (FP)-Growth algorithms are two of the most popular association analysis algorithms.

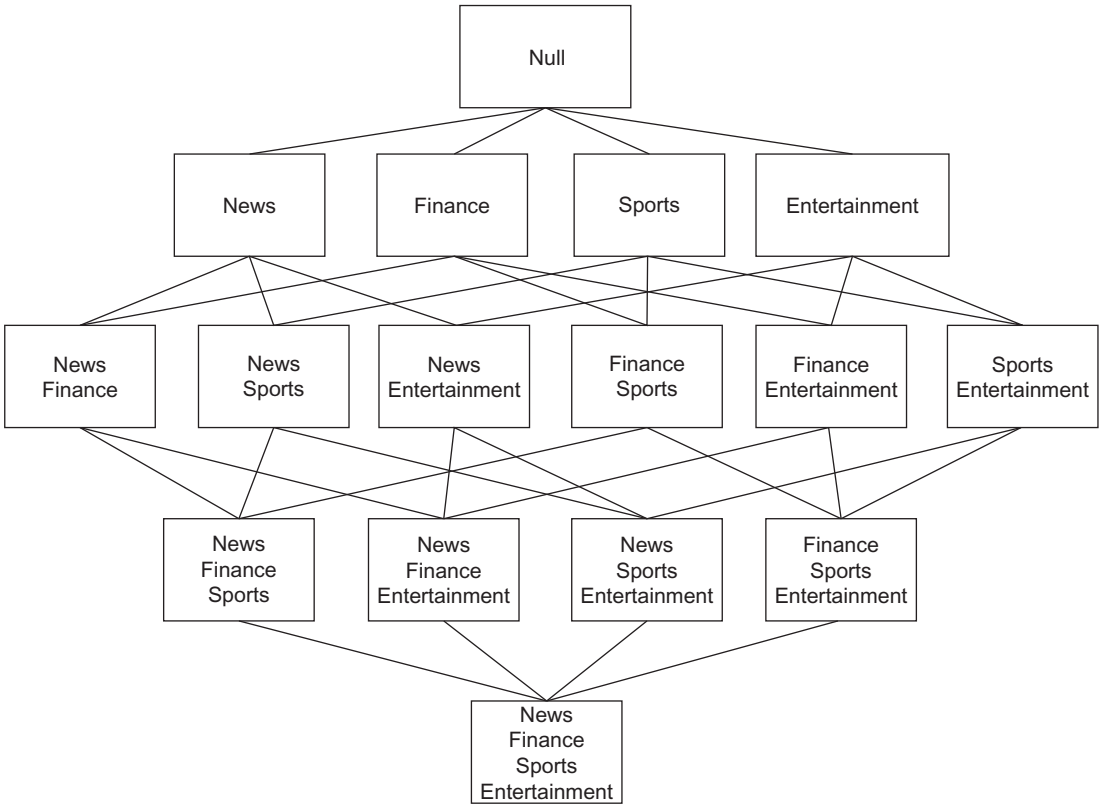


FIGURE 6.1
Itemset tree.

6.2 APRIORI ALGORITHM

All association rule algorithms should efficiently find the frequent itemsets from the universe of all the possible itemsets. The Apriori algorithm leverages some simple logical principles on the lattice itemsets to reduce the number of itemsets to be tested for the support measure (Agrawal & Srikant, 1994). The Apriori principles states that "If an itemset is frequent, then all its subset items will be frequent." (Tan et al, 2005). The itemset is "frequent" if the support for the itemset is more than that of the support threshold.

For example, if the itemset {News, Finance, Sports} from the dataset shown in Table 6.2 is a frequent itemset, that is, its support measure (0.33) is higher than the threshold support measure k (say, 0.25), then all of its subset items or itemsets will be frequent itemsets. Subset itemsets will have a support measure higher than or equal to the parent itemset. Fig. 6.2 shows the

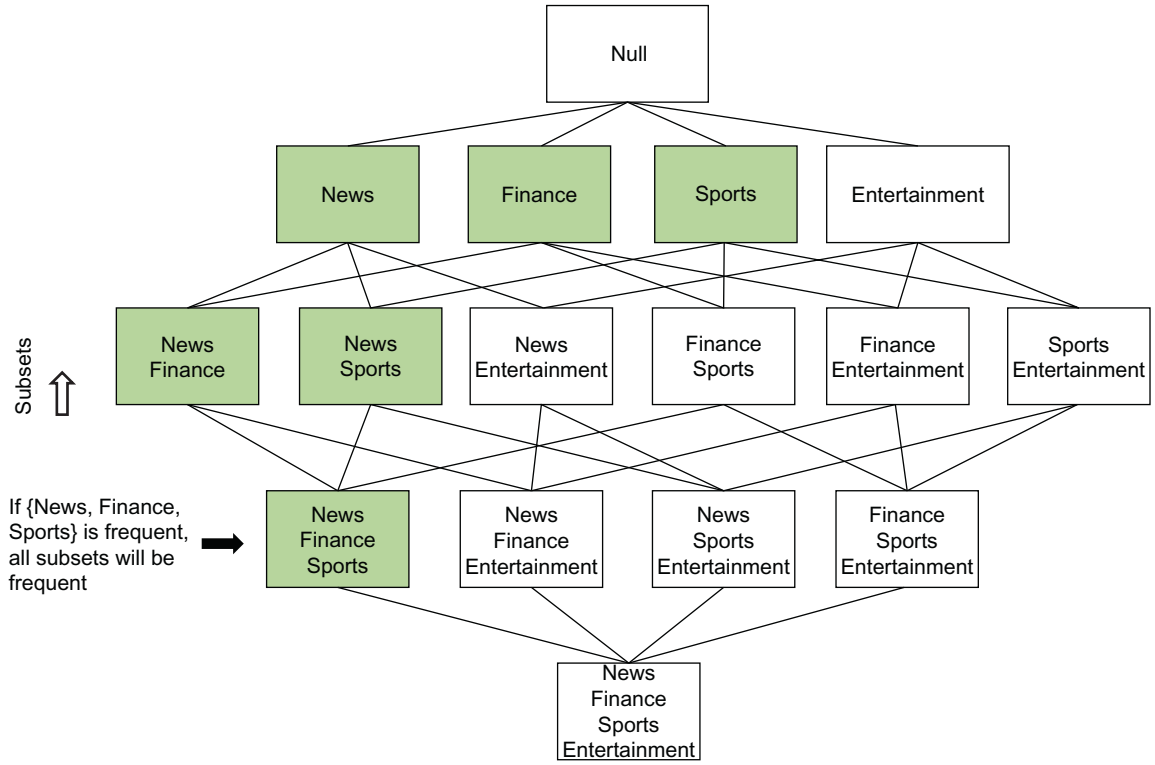


FIGURE 6.2
Frequent itemsets using Apriori principle.

application of the Apriori principle in a lattice. The support measures of the subset itemsets for {News, Finance, Sports} are

Support {News, Finance, Sports} = 0.33 (above threshold support)
Support {News, Finance} = 0.66
Support {News, Sports} = 0.33
Support {News} = 0.83
Support {Sports} = 0.33
Support {Finance} = 0.66

Conversely, if the itemset is infrequent, then all its *supersets* will be infrequent. In this example, support of Entertainment is 0.16, and the support of all the supersets that contain Entertainment as an item will be less than or equal to 0.16, which is infrequent when considering the support threshold of 0.25. Superset exclusion of an infrequent item is shown in Fig. 6.3.

The Apriori principle is helpful because not all itemsets have to be considered for a support calculation and tested for the support threshold; hence,

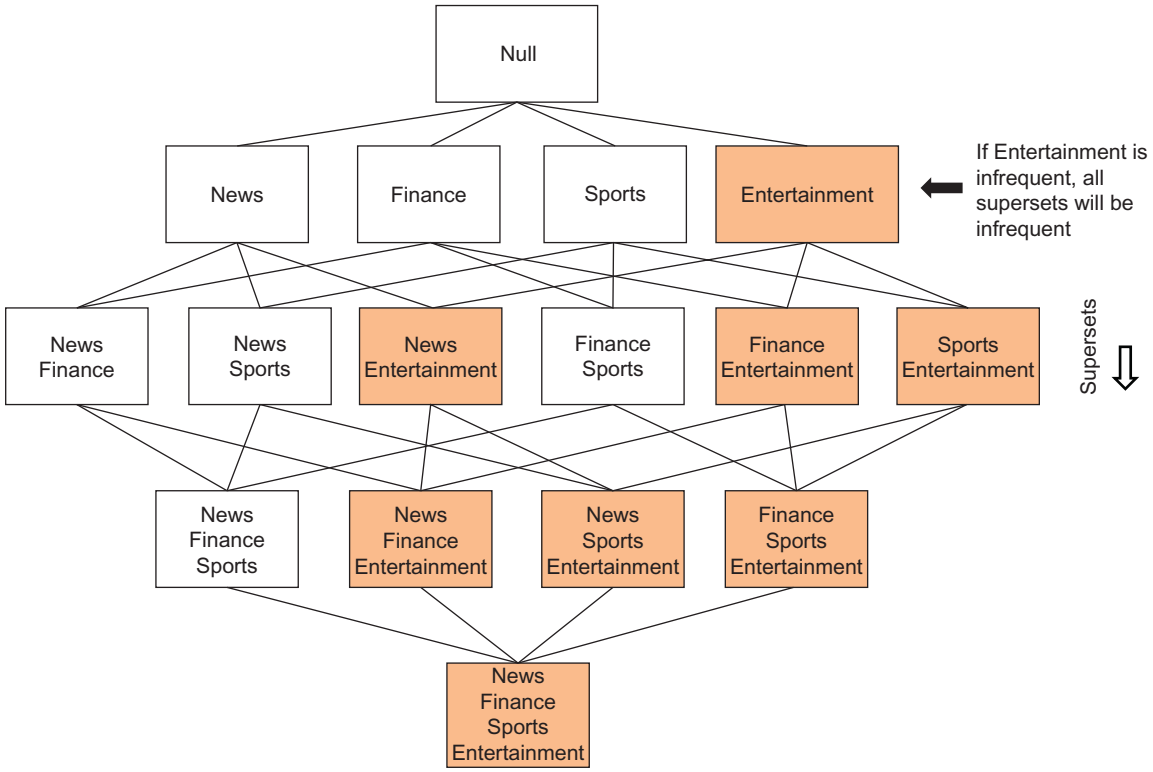


FIGURE 6.3
Frequent itemsets using Apriori principle: exclusion.

generation of the frequent itemsets can be handled efficiently by eliminating a bunch of itemsets that have an infrequent item or itemsets (Bodon, 2005).

6.2.1 How it Works

Consider the dataset shown in Table 6.3, which is the condensed version of the prior example set discussed. In this dataset there are six transactions. If the support threshold is assumed to be 0.25, then all items are expected to appear in at least two out of six transactions.

Frequent Itemset Generation

The support count and support for all itemset(s) can now be calculated. *Support count* is the absolute count of the transactions and support is the ratio of support count to total transaction count. Any one itemset below the threshold support count (which is 2 in this example) can be eliminated from further processing. Table 6.4 shows the support count and support calculation for each

Table 6.3 Clickstream Dataset: Condensed Version

Session	News	Finance	Entertainment	Sports
1	1	1	0	0
2	1	1	0	0
3	1	1	0	1
4	0	0	0	0
5	1	1	0	1
6	1	0	1	0

Table 6.4 Frequent Itemset Support Calculation

Item	Support Count	Support
{News}	5	0.83
{Finance}	4	0.67
{Entertainment}	1	0.17
{Sports}	2	0.33
Two-Itemsets	**Support Count**	**Support**
{News, Finance}	4	0.67
{News, Sports}	2	0.33
{Finance, Sports}	2	0.33
Three-Itemsets	**Support Count**	**Support**
{News, Finance, Sports}	2	0.33

item. Since {Entertainment} has a support count less than the threshold, it can be eliminated for the next iteration of itemset generation. The next step is generating possible two-itemset generations for {News}, {Finance}, and {Sports}, which yield three two-itemsets. If the {Entertainment} itemset is not eliminated, six two-itemsets would be obtained. Fig. 6.4 shows the visual representation of the itemsets with elimination of the {Entertainment} item.

This process is continued until all n-itemsets are considered from previous sets. At the end, there are seven frequent itemsets passing the support threshold. The total possible number of itemsets is 15 ($=2^4-1$). By eliminating {Entertainment} in the first step, seven additional itemsets do not have to be generated that would not pass the support threshold anyway (Witten & Frank, 2005).

Rule Generation

Once the frequent itemsets are generated, the next step in association analysis is generating useful rules which have a clear antecedent (premise) and consequent (conclusion), in the format of the rule:

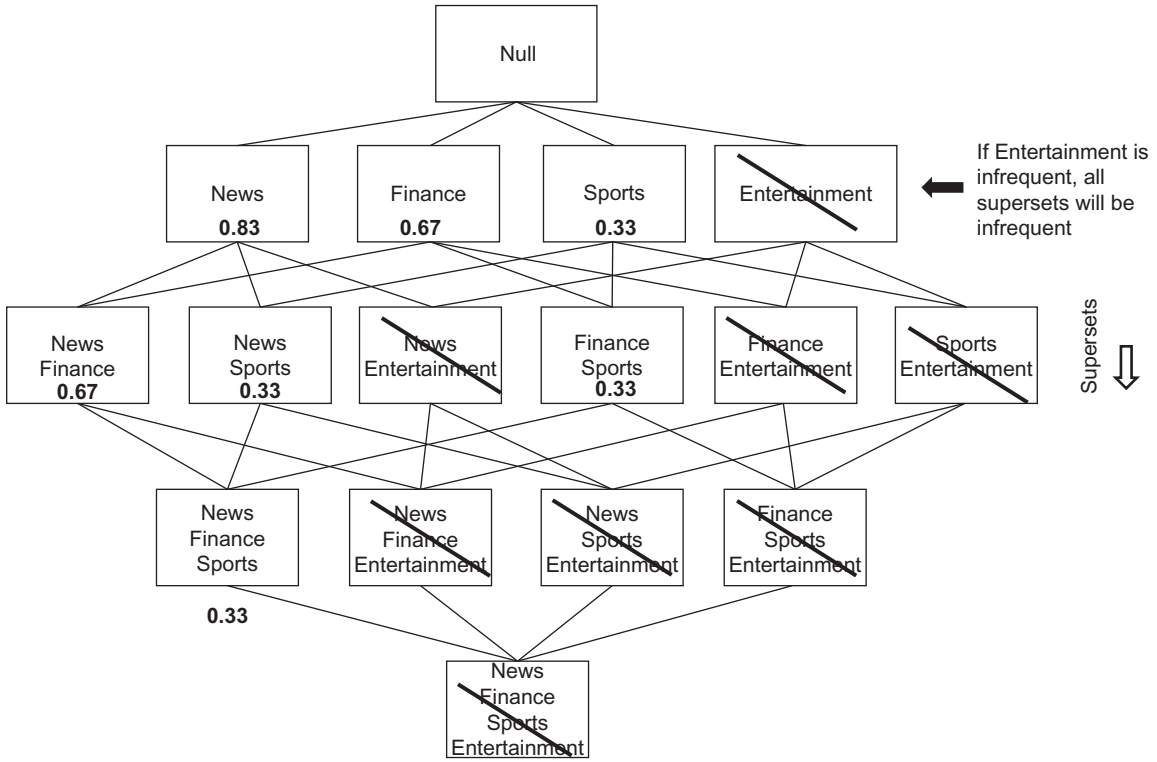


FIGURE 6.4
Frequent itemset with support.

{Item A} → {Item B}
The usefulness of the rule can be approximated by an objective measure of interest such as confidence, conviction, or lift. Confidence for the rule is calculated by the support scores of the individual items as given in Eq. (6.1). Each frequent itemset of n items can generate $2^n - 2$ rules. For example {News, Sports, Finance} can generate rules with confidence scores as follows:

Rules and confidence scores

{News, Sports} → {Finance} − 0.33/0.33 = 1.0
{News, Finance} → {Sports} − 0.33/0.67 = 0.5
{Sports, Finance} → {News} − 0.33/0.33 = 1.0
{News} → {Sports, Finance} − 0.33/0.83 = 0.4
{Sports} → {News, Finance} − 0.33/0.33 = 1.0
{Finance} → {News, Sports} − 0.33/0.67 = 0.5

Since all the support scores have already been calculated in the itemset generation step, there is no need for another set of computations for calculating

confidence. However, it is possible to prune potentially low confidence rules using the same Apriori method. For a given frequent itemset {News, Finance, Sports}, if the rule {News, Finance} → {Sports} is a low confidence rule, then it can be concluded that any rules within the subset of the antecedent will be a low confidence rule. Hence, all the rules like {News} → {Sports, Finance} and {Finance} → {News, Sports} can be discarded, which are in the subsets of the antecedent of the rule. The reason is that all three rules have the same numerator in the confidence score calculation [Eq. (6.1)], which is 0.33. The denominator calculation depends on the support of the antecedent. Since the support of a subset is always greater or equal to the set, it can be concluded that all further rules within a subset of an itemset in the premises will be a low confidence rule, and hence, can be ignored.

All the rules passing a particular confidence threshold are considered for output along with both support and confidence measures. These rules should be further evaluated for rational validity to determine if a useful relationship was uncovered, if there was an occurrence by chance, or if the rule confirms a known intuitive relationship.

6.3 FREQUENT PATTERN-GROWTH ALGORITHM

The FP-Growth algorithm provides an alternative way of calculating a frequent itemset by compressing the transaction records using a special graph data structure called *FP-Tree*. FP-Tree can be thought of as a transformation of the dataset into graph format. Rather than the generate and test approach used in the Apriori algorithm, FP-Growth first generates the FP-Tree and uses this compressed tree to generate the frequent itemsets. The efficiency of the FP-Growth algorithm depends on how much compression can be achieved in generating the FP-Tree (Han, Pei, & Yin, 2000).

6.3.1 How it Works

Consider the dataset shown in Table 6.5 containing six transactions of four items—news, finance, sports, and entertainment. To visually represent this dataset in a tree diagram (Fig. 6.6), the list of transactions need to be transformed into a tree map, preserving all the information and representing the *frequent paths*. Here is a break-down of the FP-Tree for this dataset step by step.

1. The first step is to sort all the items in each transaction in descending order of frequency (or support count). For example, News is the most frequent item and Sports is the least frequent item in the transaction, based on the data in Table 6.5. The third transaction of {Sports, News, Finance} has to be rearranged to {News, Finance, Sports}. This will help to simplify mapping frequent paths in later steps.

Table 6.5 Transactions List: Session and Items

Session	Items
1	{News, Finance}
2	{News, Finance}
3	{News, Finance, Sports}
4	{Sports}
5	{News, Finance, Sports}
6	{News, Entertainment}

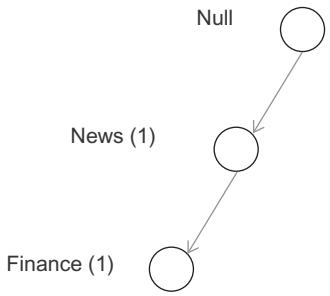


FIGURE 6.5
FP-Tree: Transaction 1.

2. Once the items within a transaction are rearranged, the transaction can now be mapped to the FP-Tree. Starting with a null node, the first transaction {News, Finance} can be represented by Fig. 6.5. The number within the parenthesis next to the item name is the number of transactions following the path.
3. Since the second transaction {News, Finance} is the same as the first one, it follows the same path as the first one. In this case, the numbers can simply be incremented.
4. The third transaction contains {News, Finance, Sports}. The tree is now extended to include Sports and the item path count is incremented (Fig. 6.6).
5. The fourth transaction only contains the {Sports} item. Since Sports is not preceded by News and Finance, a new path should be created from the null item and the item count should be noted. This node for Sports is different from the Sports node next to Finance (the latter co-occurs with News and Finance). However, since both nodes indicate the same item, they should be linked by a dotted line.
6. This process is continued until all the transactions are scanned. All of the transaction records can be now represented by a compact FP-Tree (Fig. 6.7).

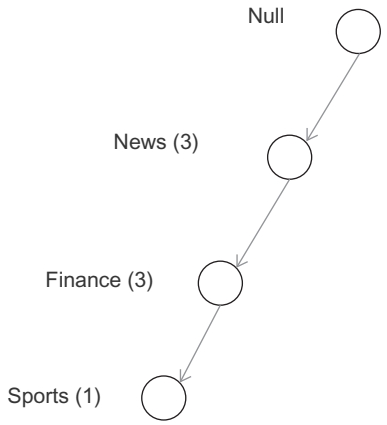


FIGURE 6.6
FP-Tree: Transactions 1—3.

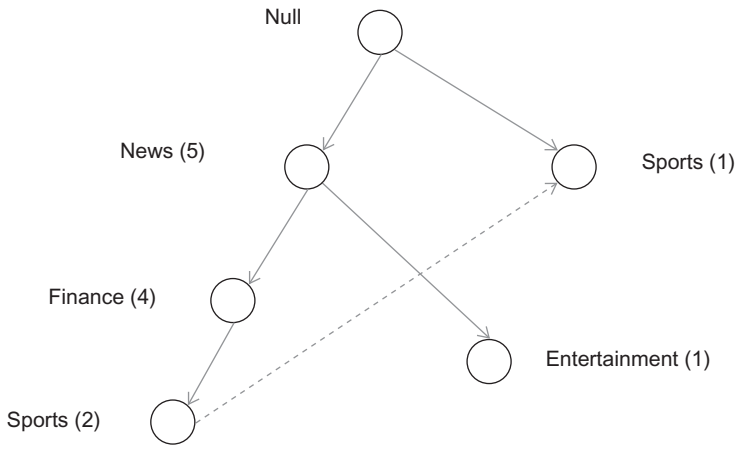


FIGURE 6.7
FP-Tree: Transactions 1—6.

The compression of the FP-Tree depends on how frequently a path occurs within a given transaction set. Since the key objective of association analysis is to identify these common paths, the datasets used from this analysis contain many frequent paths. In the worst case, all transactions contain unique itemset paths and there wouldn't be any compression. In that case the rule generation itself would be less meaningful for association analysis.

Frequent Itemset Generation

Once the transaction set is expressed by a compact FP-Tree, the most frequent itemset can be generated from the FP-Tree effectively. To generate the frequent itemset, the FP-Growth algorithm adopts a bottom-up approach of generating all the itemsets starting with the least frequent items. Since the structure of the tree is ordered by the support count, the least frequent items can be found in the leaves of the tree. In Fig. 6.8, the least frequent items are {Entertainment} and {Sports}, because the support count is just one transaction. If {Entertainment} is indeed a frequent item, because the support exceeds the threshold, the algorithm will find all the itemsets ending with entertainment, like {Entertainment} and {News, Entertainment}, by following the path from the bottom-up. Since the support counts are mapped to the nodes, calculating the support for {News, Entertainment} will be instant. If {Entertainment} is not frequent, the algorithm skips the item and goes with the next item, {Sports}, and finds all possible itemsets ending with sports: {Sports}, {Finance, Sports}, {News, Finance, Sports}.

Finding the entire itemset ending with a particular item number is actually made possible by generating a prefix path and conditional FP-Tree for an item, as shown in Fig. 6.9. The prefix path of an item is a subtree with only paths that contain the item of interest. A conditional FP-Tree for an item, say {Sports}, is similar to the FP-Tree, but with the {Sports} item removed. Based on the conditional FP-Tree, the algorithm repeats the process of finding leaf nodes. Since leaf nodes of the sports conditional tree co-exist with {Sports}, the algorithm finds the association with finance and generates {Finance, Sports}.

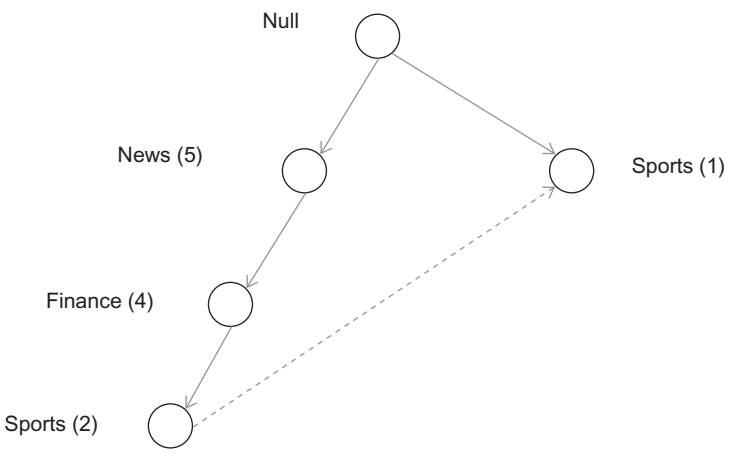


FIGURE 6.8
Trimmed FP-Tree.

Rule generation in the FP-Growth algorithm is very similar to the Apriori algorithm. Since the intent is to find frequently occurring items, by definition, many of the transactions should have essentially the same path. Hence, in many practical applications the compaction ratio is very high. In those scenarios, the FP-Growth algorithm provides efficient results. Since the FP-Growth algorithm uses graphs to map the relationship between frequent items, it has found applications beyond association analysis. It is now applied in research as a preprocessing phase for document clustering, text mining, and sentiment analysis (Akbar & Angryk, 2008). However, in spite of the execution differences, both the FP-Growth and Apriori algorithms yield similar results. Rule generation from the frequent itemsets is similar to the Apriori algorithm. Even though the concepts and explanations include analyzing graphs and subgraphs, FP-Growth algorithms can be easily ported to programming languages, particularly to SQL and PL/SQL programs on top of relational databases, where the transactions are usually stored (Shang, Sattler, & Geist, 2004).

6.3.2 How to Implement

The retrieval of association rules from a dataset is implemented through the FP-Growth algorithm in RapidMiner. Since the modeling parameters and the results for most of the association algorithms are the same, the FP-Growth algorithm will be used to observe the inputs, process, and the result of an association analysis implementation.

Step 1: Data Preparation

The Association analysis process expects transactions to be in a particular format. The input grid should have binominal (true or false) data with items in the columns and each transaction as a row. If the dataset contains transaction IDs or session IDs, they can either be ignored or tagged as a special attribute in RapidMiner. Datasets in any other format have to be converted to this

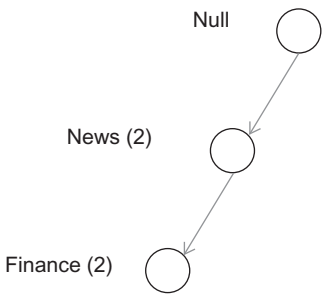


FIGURE 6.9
Conditional FP-Tree.

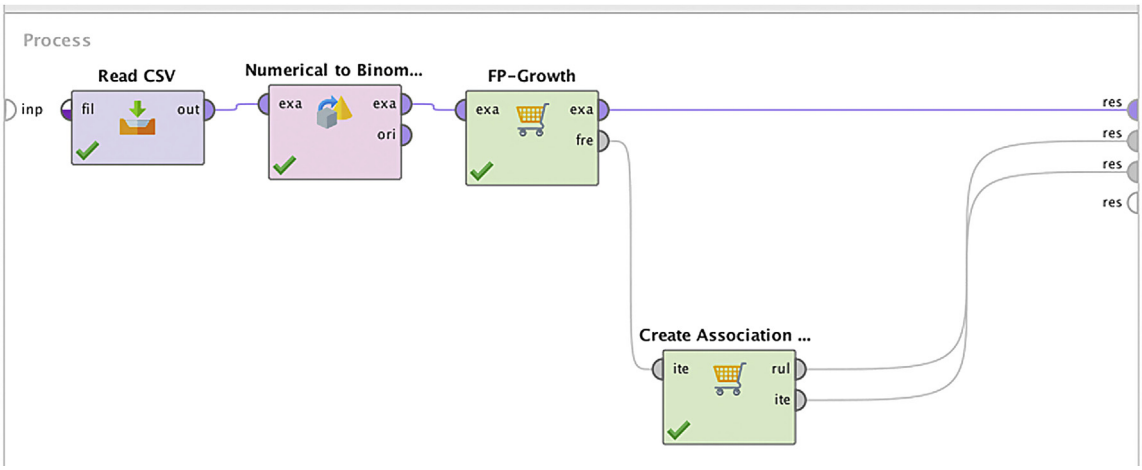


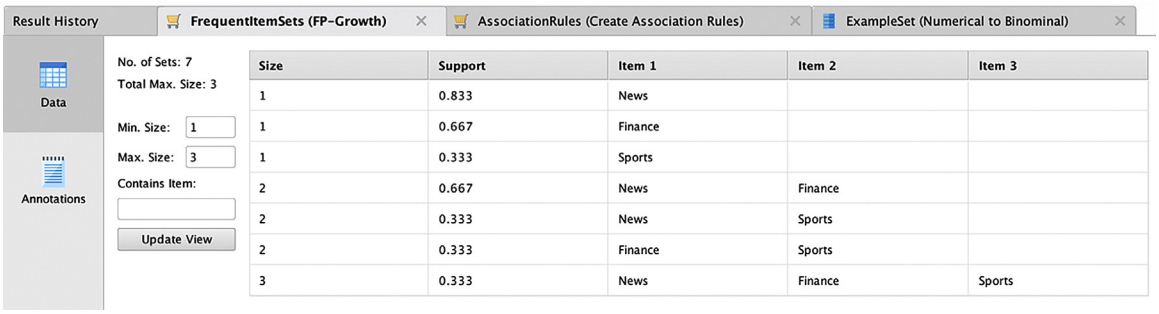
FIGURE 6.10
Data science process for FP-Growth algorithm.

transactional format using data transformation operators. In this example, the data shown in Table 6.3 has been used, with a session ID on each row and content accessed in the columns, indicated by 1 and 0. This integer format has to be converted to a binomial format by a *numerical to binominal* operator. The output of *numerical to binominal* is then connected to the *FP-Growth* operator to generate frequent itemsets. The dataset and RapidMiner process for association analysis can be accessed from the companion site of the book at www.IntroDataScience.com. Fig. 6.10 shows the RapidMiner process of association analysis with the FP-Growth algorithm.

Step 2: Modeling Operator and Parameters

The *FP-Growth* operator in RapidMiner generates all the frequent itemsets from the input dataset meeting a certain parameter criterion. The modeling operator is available at Modeling > Association and itemset Mining folder. This operator can work in two modes, one with a specified number of high support itemsets (default) and the other with minimum support criteria. These parameters can be set in this operator, thereby, affecting the behavior of the model:

- *Min Support:* Threshold for support measure. All the frequent itemsets passing this threshold will be provided in the output.
- *Max Items:* Maximum number of items in an itemset. Specifying this parameter limits too many items in an itemset.
- *Must Contain:* Regular expression to filter itemsets to contain specified items. Use this option to filter out items.



Size	Support	Item 1	Item 2	Item 3
1	0.833	News		
1	0.667	Finance		
1	0.333	Sports		
2	0.667	News	Finance	
2	0.333	News	Sports	
2	0.333	Finance	Sports	
3	0.333	News	Finance	Sports

FIGURE 6.11
Frequent itemset output.

- *Find Minimum Number of Itemsets:* This option allows the *FP-Growth* operator to lower the support threshold, if fewer itemsets are generated with the given threshold. The support threshold is decreased by 20% in each retry.
  - *Min Number of Itemsets:* Value of minimum number of itemsets to be generated.
  - *Max Number of Retries:* Number of retries allowed in achieving minimum itemsets

In this example, *Min Support* is set to 0.25. The result of the *FP-Growth* operator is the set of itemsets generated, which can be viewed in the results page. The reporting options include filtering based on the number of items and sorting based on the support threshold. Fig. 6.11 shows the output of frequent itemsets operator where all possible itemsets with support higher than the threshold can be seen.

Step 3: Create Association Rules

The next step in association analysis is generation of the most interesting rules from the frequent itemsets created from the *FP-Growth* operator. The *Create Association Rules* operator generates relevant rules from frequent itemsets. The interest measure of the rule can be specified by providing the correct interest criterion based on the dataset under investigation. The input of the *Create Association Rules* operator are frequent itemsets from the *FP-Growth* operator and the output generates all the association rules meeting the interest criterion. These parameters govern the functionality of this operator:

- *Criterion:* Used to select the interest measure to filter the association rules. All other parameters change based on the criterion selection. Confidence, lift, and conviction are commonly used interest criterion.

- *Min Criterion Value:* Specifies the threshold. Rules not meeting the thresholds are discarded.
- The *Gain theta* and *Laplace* parameters are the values specified when using Gain and Laplace parameters for the interest measure.

In this example process, we are using confidence as the criterion and a confidence value of 0.5. Fig. 6.10 shows the completed RapidMiner process for association analysis. The process can be saved and executed.

Step 4: Interpreting the Results

The filtered association analysis rules extracted from the input transactions can be viewed in the results window (Fig. 6.12). The listed association rules are in a table with columns including the premise and conclusion of the rule, as well as the support, confidence, gain, lift, and conviction of the rule. The interactive control window on the left-hand side of the screen allows the users to filter the processed rules to contain the selected item and there is a slide bar to increase the confidence or criterion threshold, thereby, showing fewer rules.

The main purpose of association analysis is to understand the relationship between items. Since the items take the role of both premise and conclusion, a visual representation of relationships between all the items, through a rule,

AssociationRules (Create Association Rules)

No.	Premises	Conclusion	Support	Confidence	LaPlace	Gain	p-s	Lift	Convic...
1	Finance	Sports	0.333	0.500	0.800	-1	0.111	1.500	1.333
2	Finance	News, Sports	0.333	0.500	0.800	-1	0.111	1.500	1.333
3	News, Finance	Sports	0.333	0.500	0.800	-1	0.111	1.500	1.333
4	News	Finance	0.667	0.800	0.909	-1	0.111	1.200	1.667
5	Finance	News	0.667	1	1	-0.667	0.111	1.200	∞
6	Sports	News	0.333	1	1	-0.333	0.056	1.200	∞
7	Sports	Finance	0.333	1	1	-0.333	0.111	1.500	∞
8	Sports	News, Finance	0.333	1	1	-0.333	0.111	1.500	∞
9	News, Sports	Finance	0.333	1	1	-0.333	0.111	1.500	∞
10	Finance, Sports	News	0.333	1	1	-0.333	0.056	1.200	∞

FIGURE 6.12
Association rules output.

can help comprehend the analysis. Fig. 6.13 shows the rules in text format and by interconnected graph format through the results window, for selected items. Fig. 6.13B shows the items selected, connected with the rules through arrows. The incoming item to a rule is the premise of the rule and the outgoing item is the conclusion of the association rule.

(A)

AssociationRules

```
Association Rules
[Finance] --> [Sports] (confidence: 0.500)
[Finance] --> [News, Sports] (confidence: 0.500)
[News, Finance] --> [Sports] (confidence: 0.500)
[News] --> [Finance] (confidence: 0.800)
[Finance] --> [News] (confidence: 1.000)
[Sports] --> [News] (confidence: 1.000)
[Sports] --> [Finance] (confidence: 1.000)
[Sports] --> [News, Finance] (confidence: 1.000)
[News, Sports] --> [Finance] (confidence: 1.000)
[Finance, Sports] --> [News] (confidence: 1.000)
```

(B)

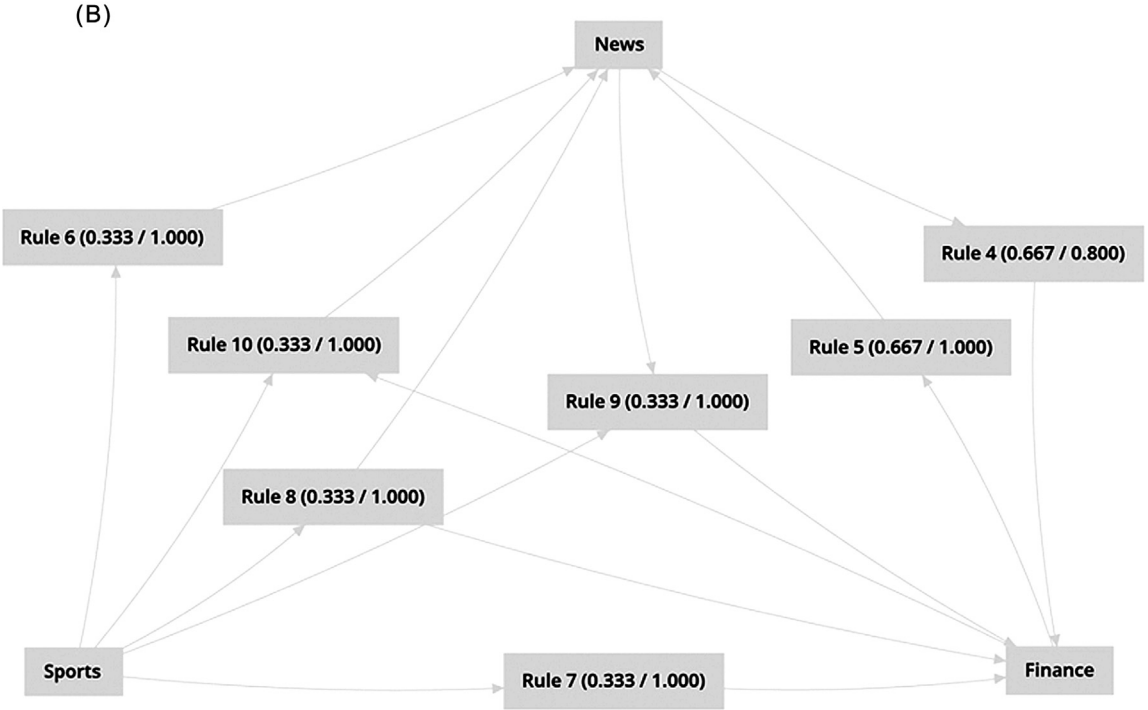


FIGURE 6.13
Association rules output (A) text view and (B) graph view.

6.4 CONCLUSION

Association rules analysis has gained popularity in the last two decades particularly in retail, online cross selling, recommendation engines, text analysis, document analysis, and web analysis. Typically, a commercial data science tool offers association analysis in its tool package. Though there may be a variation in how the algorithm is implemented in each commercial package, the framework of generating a frequent itemset using a support threshold and generating rules from the itemsets using an interest criterion is the same. Applications that involve large amount of items and real-time decision making, demand new approaches with efficient and scalable association analysis (Zaki, 2000). Association analysis is also one of the prevalent algorithms that is applied to information stored using big data technologies, data streams, and large databases (Tanbeer, Ahmed, Jeong, & Lee, 2008).

References

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In *SIGMOD '93 proceedings of the 1993 ACM SIGMOD international conference on management of data* (pp. 207–216).

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *The international conference on very large databases*, 487–499.

Akbar, M., & Angryk, R. (2008). Frequent pattern-growth approach for document organization. In *Proceeding of the 2nd international workshop on Ontologies and information systems for the semantic web, ACM* (pp. 77–82). Available from http://dl.acm.org/citation.cfm?id = 1458496.

Bodon, F. (2005). A trie-based APRIORI implementation for mining frequent item sequences. In *Proceedings of the 1st international workshop on open source data science frequent pattern mining implementations – OSDM '05* (pp. 56–65). http://dx.doi.org/10.1145/1133905.1133913.

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. In *SIGMOD '00 proceedings of the 2000 ACM SIGMOD international conference on management of data* (pp. 1–12).

Power, D.J. (2002, Nov 10). *DSS News*. Retrieved Jan 21, 2014, from Desicion Support Systems (DSS). Retrieved Jan 21, 2014, from Desicion Support Systems (DSS), http://www.dssresources.com/newsletters/66.php.

Shang, X., Sattler, K.U., Geist, I. (2004). SQL based frequent pattern mining without candidate generation. In *2004 ACM symposium on applied computing – Poster Abstract* (pp. 618–619).

Tan, P.-N., Steinbach, M., & Kumar, V. (2005). *Association analysis: Basic concepts and algorithms. Introduction to data mining* (pp. 327–404). Boston, MA: Addison Wesley.

Tanbeer, S.K., Ahmed, C.F., Jeong, B.-S., Lee, Y.-K. (2008). Efficient frequent pattern mining over data streams. In *Proceeding of the 17th ACM conference on information and knowledge mining – CIKM '08* (Vol. 1, pp. 1447–1448). http://dx.doi.org/10.1145/1458082.1458326.

Witten, I. H., & Frank, E. (2005). *Algorithms: The basic methods: Mining association rules. Data science: Practical machine learning tools and techniques* (pp. 112–118). San Francisco, CA: Morgan Kaufmann.

Zaki, M. Jk (2000). Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering, 12*(3), 372–390. Available from https://doi.org/10.1109/69.846291.