Lecture 10 - Association Rules

Adnan Ferdous Ashrafi

Stamford University Bangladesh

Table of Contents

AFFINITY ANALYSIS AND MARKET BASKET ANALYSIS- Definition

Affinity analysis is the study of attributes or characteristics that "go together." Methods for affinity analysis, also known as *market basket analysis*, seek to uncover associations among these attributes; that is, it seeks to uncover rules for quantifying the relationship between two or more attributes. Association rules take the form "If antecedent, then consequent," along with a measure of the *support* and *confidence* associated with the rule.

For example, a particular supermarket may find that of the 1000 customers shopping on a Thursday night, 200 bought a pen, and of the 200 who bought a pen, 50 bought paper. Thus, the association rule would be: "If buy pen, then buy paper," with a support of 50/1000 = 5% and a confidence of 50/200 = 25%.

Examples of association tasks

Examples of association tasks in business and research include:

- Investigating the proportion of subscribers to your company's cell phone plan that respond positively to an offer of a service upgrade
- Examining the proportion of children whose parents read to them who are themselves good readers
- Predicting degradation in telecommunications networks
- Finding out which items in a supermarket are purchased together, and which items are never purchased together
- Determining the proportion of cases in which a new drug will exhibit dangerous side effects

Concerns for Association analysis

What types of algorithms can we apply to mine association rules from a particular data set? The daunting problem that awaits any such algorithm is the curse of dimensionality: The number of possible association rules grows exponentially in the number of attributes. Specifically, if there are k attributes, we limit ourselves to binary attributes, we account only for the positive cases (e.g., buy pen = yes), there are on the order of $k \cdot 2^{k-1}$ possible association rules.

Concerns for Association analysis

Consider that a typical application for association rules is market basket analysis and that there may be thousands of binary attributes *(buy pen? buy paper? buy milk? buy bread? etc.)*, the search problem appears at first glance to be utterly hopeless.

For example, suppose that a tiny convenience store has only 100 different items, and a customer could either buy or not buy any combination of those 100 items. Then there are $100 \cdot 2^{99} \simeq 6.4 \times 10^{31}$ possible association rules that await your intrepid search algorithm.

Apriori Algorithm

The *a priori* algorithm [1] for mining association rules, however, takes advantage of structure within the rules themselves to reduce the search problem to a more manageable size. Before we examine the a priori algorithm, however, let us consider some basic concepts and notation for association rule mining.

We begin with a simple example.

Apriori Algorithm- Example

Suppose that a local farmer has set up a roadside vegetable stand and is offering the following items for sale: asparagus, beans, broccoli, corn, green peppers, squash, tomatoes. Denote this set of items as I . One by one, customers pull over, pick up a basket, and purchase various combinations of these items, subsets of I . (For our purposes, we don't keep track of how much of each item is purchased, just whether or not that particular item is purchased.) Suppose Table 1 lists the transactions made during one fine fall afternoon at this roadside vegetable stand.

Apriori Algorithm- Example

Table 1: Transactions Made at the Roadside Vegetable Stand

Transaction	Items Purchased
1	Broccoli, green peppers, corn
2	Asparagus, squash, corn
3	Corn, tomatoes, beans, squash
4	Green peppers, corn, tomatoes, beans
5	Beans, asparagus, broccoli
6	Squash, asparagus, beans, tomatoes
7	Tomatoes, corn
8	Broccoli, tomatoes, green peppers
9	Squash, asparagus, beans
10	Beans, corn
11	Green peppers, broccoli, beans, squash
12	Asparagus, beans, squash
13	Squash, corn, asparagus, beans
14	Corn, green peppers, tomatoes, beans, broccoli

Data Representation for Market Basket Analysis

There are two principal methods of representing this type of market basket data: using either the transactional data format or the tabular data format. The *transactional data format* requires only two fields, an *ID* field and a content field, with each record representing a single item only.

For example, the data in Table 1 could be represented using transactional data format as shown in Table 2. In the tabular data format, each record represents a separate transaction, with as many 0/1 flag fields as there are items. The data from Table 2 could be represented using the tabular data format, as shown in Figure 1.

Data Representation for Market Basket Analysis

Table 2: Transactional Data Format for the Roadside Vegetable Stand Data

Transaction	Items Purchased
1	Broccoli, green peppers, corn
1	Broccoli, green peppers, cornn
1	Broccoli, green peppers, corn
4	Green peppers, corn, tomatoes, beans
5	Asparagus
6	Squash
7	Corn
8	Corn
9	Tomatoes
...	...

SUPPORT, CONFIDENCE, FREQUENT ITEMSETS, AND THE A PRIORI PROPERTY- Definition

Let D be the set of transactions represented in Table 1, where each transaction T in D represents a set of items contained in I . Suppose that we have a particular set of items A (e.g., beans and squash), and another set of items B (e.g., asparagus).

Then an association rule takes the form if A, then B (i.e., $A \Rightarrow B$), where the antecedent A and the consequent B are proper subsets of I, and A and B are mutually exclusive. This definition would exclude, for example, trivial rules such as if beans and squash, then beans.

Definition

Transaction	Asparagus	Beans	Broccoli	Corn	Green Peppers	Squash	Tomatoes
1	0	0	1	1	1	0	0
2	1	0	0	1	0	1	0
3	0	1	0	1	0	1	1
4	0	1	0	1	1	0	1
5	1	1	1	0	0	0	0
6	1	1	0	0	0	1	1
7	0	0	0	1	0	0	1
8	0	0	1	0	1	0	1
9	1	1	0	0	0	1	0
10	0	1	0	1	0	0	0
11	0	1	1	0	1	1	0
12	1	1	0	0	0	1	0
13	1	1	0	1	0	1	0
14	0	1	1	1	1	0	1

Figure 1: Tabular Data Format for the Roadside Vegetable Stand Data

Support

The support s for a particular association rule $A \Rightarrow B$ is the proportion of transactions in D that contain both A and B. That is,

$$\text{support} = P(A \cap B) = \frac{\text{number of transactions containing both A and B}}{\text{total number of transactions}}$$

Confidence

The confidence c of the association rule $A \Rightarrow B$ is a measure of the accuracy of the rule, as determined by the percentage of transactions in D containing A that also contain B. In other words,

$$\begin{aligned} \text{confidence} = P(B|A) &= \frac{P(A \cap B)}{P(A)} \\ &= \frac{\text{number of transactions containing both A and B}}{\text{number of transactions containing A}} \end{aligned}$$

Strong Rules

Analysts may prefer rules that have either high support or high confidence, and usually both. *Strong rules* are those that meet or surpass certain minimum support and confidence criteria.

For example, an analyst interested in finding which supermarket items are purchased together may set a minimum support level of 20% and a minimum confidence level of 70%. On the other hand, a fraud detection analyst or a terrorism detection analyst would need to reduce the minimum support level to 1% or less, since comparatively few transactions are either fraudulent or terror-related.

Itemset

An itemset is a set of items contained in I , and a $k-itemset$ is an itemset containing k items. For example, {beans, squash} is a 2-itemset, and {broccoli, green peppers, corn} is a 3-itemset, each from the vegetable stand set I. The itemset frequency is simply the number of transactions that contain the particular itemset.

A frequent itemset is an itemset that occurs at least a certain minimum number of times, having itemset frequency $\geq \phi$. For example, suppose that we set $\phi = 4$. Then itemsets that occur more than four times are said to be frequent. We denote the set of frequent k-itemsets as F_k.

Mining Association Rules

Mining Association Rules

The mining of association rules from large databases is a two-steps process:

1. Find all frequent itemsets; that is, find all itemsets with frequency $\geq \phi$.
2. From the frequent itemsets, generate association rules satisfying the minimum support and confidence conditions.

A-priori Property

The a priori algorithm takes advantage of the a priori property to shrink the search space. The a priori property states that if an itemset Z is not frequent, then adding another item A to the itemset Z will not make Z more frequent. That is, if Z is not frequent, $Z \cup A$ will not be frequent. In fact, no superset of Z (itemset containing Z) will be frequent. This helpful property reduces significantly the search space for the a priori algorithm.

A-priori Property

If an itemset Z is not frequent then for any item A, $Z \cup A$ will not be frequent.

HOW DOES THE A PRIORI ALGORITHM WORK- GENERATING FREQUENT ITEMSETS- 1-itemsets

Consider the set of transactions D represented in Table 1. How would the a priori algorithm mine association rules from this data set?

Let $\phi = 4$, so that an itemset is frequent if it occurs four or more times in D. We first find F_1 , the frequent 1-itemsets, which represent simply the individual vegetable items themselves. To do so, we may turn to Figure 1 and take the column sums, which give us the number of transactions containing each particular vegetable. Since each sum meets or exceeds $\phi = 4$, we conclude that each 1-itemset is frequent. Thus, F_1 = {asparagus, beans, broccoli, corn, green peppers, squash, tomatoes}.

GENERATING FREQUENT ITEMSETS- 2-itemsets

Next, we turn to finding the frequent 2-itemsets. In general, to find F_k, the a priori algorithm first constructs a set C_k of candidate k-itemsets by joining F_{k-1} with itself. Then it prunes C_k using the a priori property. The itemsets in C_k that survive the pruning step then form F_k . Here, C_2 consists of all the combinations of vegetables in Figure 2.

GENERATING FREQUENT ITEMSETS

Combination	Count	Combination	Count
Asparagus, beans	5	Broccoli, corn	2
Asparagus, broccoli	1	Broccoli, green peppers	4
Asparagus, corn	2	Broccoli, squash	1
Asparagus, green peppers	0	Broccoli, tomatoes	2
Asparagus, squash	5	Corn, green peppers	3
Asparagus, tomatoes	1	Corn, squash	3
Beans, broccoli	3	Corn, tomatoes	4
Beans, corn	5	Green peppers, squash	1
Beans, green peppers	3	Green peppers, tomatoes	3
Beans, squash	6	Squash, tomatoes	2
Beans, tomatoes	4		

Figure 2: Candidate 2-ItemSets

GENERATING FREQUENT ITEMSETS- 3-itemsets

Since $\phi = 4$, we have F_2 = { {asparagus, beans}, {asparagus, squash}, {beans, corn}, and {beans, squash}, {beans, tomatoes}, {broccoli, green peppers}, {corn, tomatoes} }. Next, we use the frequent itemsets in F_2 to generate C_3 , the candidate 3-itemsets. To do so, we join F_2 with itself, where itemsets are joined if they have the first $k-1$ items in common (in alphabetical order).

For example, {asparagus, beans} and {asparagus, squash} have the first $k-1 = 1$ item in common, asparagus. Thus, they are joined into the new candidate itemset {asparagus, beans, squash}. Similarly, {beans, corn} and {beans, squash} have the first item, beans, in common, generating the candidate 3-itemset {beans, corn, squash}. Finally, candidate 3-itemsets {beans, corn, tomatoes} and {beans, squash, tomatoes} are generated in like fashion. Thus, C_3 = { {asparagus, beans, squash}, {beans, corn, squash}, {beans, corn, tomatoes}, {beans, squash, tomatoes} }.

GENERATING FREQUENT ITEMSETS- 3-itemsets

C_3 is then pruned, using the a priori property. For each itemset s in C_3, its size $k-1$ subsets are generated and examined. If any of these subsets are not frequent, s cannot be frequent and is therefore pruned. For example, let s = {asparagus, beans, squash}. The subsets of size $k-1=2$ are generated, as follows: {asparagus, beans}, {asparagus, squash}, and {beans, squash}. From Table 10.4 we see that each of these subsets is frequent and that therefore s = {asparagus, beans, squash} is not pruned.

Can you verify that s = {beans, corn, tomatoes} will also not be pruned?

GENERATING FREQUENT ITEMSETS- 3-itemsets

However, consider $s = \{\text{beans, corn, squash}\}$. The subset {corn, squash} has frequency $3 < 4 = \phi$, so that {corn, squash} is not frequent. By the a priori property, therefore, {beans, corn, squash} cannot be frequent, is therefore pruned, and does not appear in F_3. Also consider $s = \{\text{beans, squash, tomatoes}\}$. The subset {squash, tomatoes} has frequency $2 < 4 = \phi$, and hence is not frequent. Again, by the a priori property, its superset {beans, squash, tomatoes} cannot be frequent and is also pruned, not appearing in F_3.

GENERATING FREQUENT ITEMSETS- 3-itemsets

We still need to check the count for these candidate frequent itemsets. The itemset {asparagus, beans, squash} occurs four times in the transaction list, {beans, corn, tomatoes} occurs only three times. Therefore, the latter candidate itemset is also pruned, leaving us with a singleton frequent itemset in F_3: {asparagus, beans, squash}. This completes the task of finding the frequent itemsets for the vegetable stand data D.

GENERATING ASSOCIATION RULES

Next, we turn to the task of generating association rules using the frequent itemsets. This is accomplished using the following two-step process, for each frequent itemset s:

GENERATING ASSOCIATION RULES

1. First, generate all subsets of s.
2. Then, let *ss* represent a nonempty subset of *s*. Consider the association rule $R : ss \Rightarrow (s - ss)$, where $(s - ss)$ indicates the set *s* without *ss*. Generate (and output) *R* if *R* fulfills the minimum confidence requirement. Do so for every subset *ss* of *s*. Note that for simplicity, a single-item consequent is often desired.

GENERATING ASSOCIATION RULES

For example, let $s = \{\text{asparagus, beans, squash}\}$ from F_3. The proper subsets of s are {asparagus}, {beans}, {squash}, {asparagus, beans}, {asparagus, squash}, {beans, squash}. For the first association rule shown in Figure 3, we let $ss = \{\text{asparagus, beans}\}$, so that $(s - ss) = \{\text{squash}\}$.We consider the rule $R : \{\text{asparagus, beans}\} \Rightarrow \{\text{squash}\}$.

The support is the proportion of transactions in which both {asparagus, beans} and {squash} occur, which is 4 (or 28.6%) of the 14 total transactions in D.

To find the confidence, we note that {asparagus, beans} occurs in five of the 14 transactions, four of which also contain {squash}, giving us our confidence of 4/5 = 80%.

GENERATING ASSOCIATION RULES

If *Antecedent*, then *Consequent*	Support	Confidence
If buy asparagus and beans, then buy squash	$4/14 = 28.6\%$	$4/5 = 80\%$
If buy asparagus and squash, then buy beans	$4/14 = 28.6\%$	$4/5 = 80\%$
If buy beans and squash, then buy asparagus	$4/14 = 28.6\%$	$4/6 = 66.7\%$

Figure 3: Candidate Association Rules for Vegetable Stand Data: Two Antecedents

GENERATING ASSOCIATION RULES

The statistics for the second rule in Figure 3 arise similarly.

For the third rule in Figure 3, the support is still 4/14 = 28.6%, but the confidence falls to 66.7%. This is because {beans, squash} occurs in six transactions, four of which also contain {asparagus}.

Assuming that our minimum confidence criterion is set at 60% and that we desire a single consequent, we therefore have the candidate rules shown in Figure 3.

If our minimum confidence were set at 80%, the third rule would not be reported.

GENERATING ASSOCIATION RULES

Finally, we turn to single antecedent/single consequent rules. Applying the association rule generation method outlined in the box above, and using the itemsets in F_2, we may generate the candidate association rules shown in Figure 4.

To provide an overall measure of usefulness for an association rule, analysts sometimes multiply the support times the confidence. This allows the analyst to rank the rules according to a combination of prevalence and accuracy. Figure 5 provides such a list for our present data set, after first filtering the rules through a minimum confidence level of 80%.

GENERATING ASSOCIATION RULES

If *Antecedent*, then *Consequent*	Support	Confidence
If buy asparagus, then buy beans	$5/14 = 35.7\%$	$5/6 = 83.3\%$
If buy beans, then buy asparagus	$5/14 = 35.7\%$	$5/10 = 50\%$
If buy asparagus, then buy squash	$5/14 = 35.7\%$	$5/6 = 83.3\%$
If buy squash, then buy asparagus	$5/14 = 35.7\%$	$5/7 = 71.4\%$
If buy beans, then buy corn	$5/14 = 35.7\%$	$5/10 = 50\%$
If buy corn, then buy beans	$5/14 = 35.7\%$	$5/8 = 62.5\%$
If buy beans, then buy squash	$6/14 = 42.9\%$	$6/10 = 60\%$
If buy squash, then buy beans	$6/14 = 42.9\%$	$6/7 = 85.7\%$
If buy beans, then buy tomatoes	$4/14 = 28.6\%$	$4/10 = 40\%$
If buy tomatoes, then buy beans	$4/14 = 28.6\%$	$4/6 = 66.7\%$
If buy broccoli, then buy green peppers	$4/14 = 28.6\%$	$4/5 = 80\%$
If buy green peppers, then buy broccoli	$4/14 = 28.6\%$	$4/5 = 80\%$
If buy corn, then buy tomatoes	$4/14 = 28.6\%$	$4/8 = 50\%$
If buy tomatoes, then buy corn	$4/14 = 28.6\%$	$4/6 = 66.7\%$

Figure 4: Candidate Association Rules for Vegetable Stand Data: One Antecedent

GENERATING ASSOCIATION RULES

If *Antecedent*, then *Consequent*	Support	Confidence	Support × Confidence
If buy squash, then buy beans	$6/14 = 42.9\%$	$6/7 = 85.7\%$	0.3677
If buy asparagus, then buy beans	$5/14 = 35.7\%$	$5/6 = 83.3\%$	0.2974
If buy asparagus, then buy squash	$5/14 = 35.7\%$	$5/6 = 83.3\%$	0.2974
If buy broccoli, then buy green peppers	$4/14 = 28.6\%$	$4/5 = 80\%$	0.2288
If buy green peppers, then buy broccoli	$4/14 = 28.6\%$	$4/5 = 80\%$	0.2288
If buy asparagus and beans, then buy squash	$4/14 = 28.6\%$	$4/5 = 80\%$	0.2288
If buy asparagus and squash, then buy beans	$4/14 = 28.6\%$	$4/5 = 80\%$	0.2288

Figure 5: Final List of Association Rules for Vegetable Stand Data: Ranked by Support × Confidence, Minimum Confidence 80%

Inference

Armed with this knowledge, the vegetable stand entrepreneur can deploy marketing strategies that take advantage of the patterns uncovered above.

Why do these particular products co-occur in customers' market baskets?

Should the product layout be altered to make it easier for customers to purchase these products together?

Should personnel be alerted to remind customers not to forget item B when purchasing associated item A?

Further Reading

- Chapter 4.5 of Data Mining - Practical Machine Learning Tools and Techniques, Second Edition - Ian H. Witten, Eibe Frank
- Chapter 10 of DISCOVERING KNOWLEDGE IN DATA - An Introduction to Data Mining - DANIEL T. LAROSE
- Chapter 6 of Introduction to Data Mining (Second Edition) - Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar

References

[1] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules in large databases,” in *Proceedings of the 20th International Conference on Very Large Data Bases*, ser. VLDB ’94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, p. 487–499.

Thank you.
Any Questions?