Frequent Itemset and Apriori Algorithm by Nikhil Vithlani

Suppose you have records of large number of transactions at a shopping centre as follows:

Transactions	Items bought
T1	Item1, item2, item3
T2	Item1, item2
T3	Item2, item5
T4	Item1, item2, item5

Learning association rules basically means finding the items that are purchased together more frequently than others.
For example in the above table you can see Item1 and item2 are bought together frequently.

What is the use of learning association rules?

- Shopping centres use association rules to place the items next to each other so that users buy more items. If you are familiar with data mining you would know about the famous beer-diapers-Wal-Mart story. Basically Wal-Mart studied their data and found that on Friday afternoon young American males who buy diapers also tend to buy beer. So Wal-Mart placed beer next to diapers and the beer-sales went up. This is famous because no one would have predicted such a result and that's the power of data mining. You can Google for this if you are interested in further details
- Also if you are familiar with Amazon, they use association mining to recommend you the items based on the current item you are browsing/buying.
- Another application is the Google auto-complete, where after you type in a word it searches frequently associated words that user type after that particular word.

So as I said Apriori is the classic and probably the most basic algorithm to do it. Now if you search online you can easily find the pseudo-code and mathematical equations and stuff. I would like to make it more intuitive and easy, if I can.

Let's start with a non-simple example,

Transaction ID	Items Bought
T1	{Mango, Onion, Nintendo, Key-chain, Eggs, Yo-yo}
T2	{Doll, Onion, Nintendo, Key-chain, Eggs, Yo-yo}
T3	{Mango, Apple, Key-chain, Eggs}
T4	{Mango, Umbrella, Corn, Key-chain, Yo-yo}
T5	{Corn, Onion, Onion, Key-chain, Ice-cream, Eggs}

Now, we follow a simple golden rule: we say an item/itemset is frequently bought if it is bought at least 60% of times. So for here it should be bought at least 3 times.

For simplicity
M = Mango
O = Onion
And so on……

So the table becomes

Original table:

Transaction ID	Items Bought
T1	{M, O, N, K, E, Y }
T2	{D, O, N, K, E, Y }
T3	{M, A, K, E}
T4	{M, U, C, K, Y }
T5	{C, O, O, K, I, E}

Step 1: Count the number of transactions in which each item occurs, Note 'O=Onion' is bought 4 times in total, but, it occurs in just 3 transactions.

Item	No of transactions
M	3
O	3
N	2
K	5
E	4
Y	3
D	1
A	1
U	1
C	2
I	1

Step 2: Now remember we said the item is said frequently bought if it is bought at least 3 times. So in this step we remove all the items that are bought less than 3 times from the above table and we are left with

Item	Number of transactions
M	3
O	3
K	5
E	4
Y	3

This is the single items that are bought frequently. Now let's say we want to find a pair of items that are bought frequently. We continue from the above table (Table in step 2)

Step 3: We start making pairs from the first item, like MO,MK,ME,MY and then we start with the second item like OK,OE,OY. We did not do OM because we already did MO when we were making pairs with M and buying a Mango and Onion together is same as buying Onion and Mango together. After making all the pairs we get,

Item pairs
MO
MK
ME
MY
OK
OE
OY
KE
KY
EY

Step 4: Now we count how many times each pair is bought together. For example M and O is just bought together in {M,O,N,K,E,Y}
While M and K is bought together 3 times in {M,O,N,K,E,Y}, {M,A,K,E} AND {M,U,C, K, Y}

After doing that for all the pairs we get

Item Pairs	Number of transactions
MO	1
MK	3
ME	2
MY	2
OK	3
OE	3
OY	2
KE	4
KY	3
EY	2

Step 5: Golden rule to the rescue. Remove all the item pairs with number of transactions less than three and we are left with

Item Pairs	Number of transactions
MK	3
OK	3
OE	3
KE	4
KY	3

These are the pairs of items frequently bought together.
Now let's say we want to find a set of three items that are brought together.
We use the above table (table in step 5) and make a set of 3 items.

Step 6: To make the set of three items we need one more rule (it's termed as self-join),
It simply means, from the Item pairs in the above table, we find two pairs with the same first Alphabet, so we get

- OK and OE, this gives OKE
- KE and KY, this gives KEY

Then we find how many times O,K,E are bought together in the original table and same for K,E,Y and we get the following table

Item Set	Number of transactions
OKE	3
KEY	2

While we are on this, suppose you have sets of 3 items say ABC, ABD, ACD, ACE, BCD and you want to generate item sets of 4 items you look for two sets having the same first two alphabets.

- ABC and ABD -> ABCD
- ACD and ACE -> ACDE

And so on … In general you have to look for sets having just the last alphabet/item different.

Step 7: So we again apply the golden rule, that is, the item set must be bought together at least 3 times which leaves us with just OKE, Since KEY are bought together just two times.

Thus the set of three items that are bought together most frequently are O,K,E.

Let's have a conceptual understanding of Significance of Support and Confidence and how they can be used practically.
Have you ever been on amazon.com? Whenever you see an item, it shows you two sections as shown in the figure. We are trying to achieve something similar using apriori algorithm

- Frequently Bought Together
- Customers who bought this item also bought

Support:

We have already covered support, remember I said

Now, we follow a simple golden rule: we say an item/itemset is frequently bought if it is bought at least 60% of times. So for here it should be bought at least 3 times.
This means we are saying that our minimum support is 60%.

In other words, minimum support 60% says, consider an itemset as a frequent itemset only if it occurs in at least 60% of transactions. We can add those items to frequently bought together section, O,K,E in our example from part 1.

Confidence:

We can take our frequent itemset knowledge even further, by finding (deducing) association rules using the frequent itemset.
In simple words, We know that O,K,E are brought together frequently, but, what is the association between them.
To do this we create, we list all the subsets of frequently bought items (O,K,E) in our case, we get following subsets,

- {O}
- {K}
- {E}
- {O,K}
- {K,E}
- {O,E}

Now, we find the association among all the subsets

{O} => {K,E} *:* (if 'O' is bought, what is the probability that 'K' and 'E' would be bought in the same transaction) O is bought in 3 transactions and in all those transactions 'K' and 'E' are also bought, i.e. Confidence = 3/3 x 100 = 100%. We can say with 100% confidence that if O is bought K and E would also be bought, so If you have an online store like amazon.com and a user is viewing Onion, you can say, customers who bought Onion also bought Key-chain and Eggs.

On the same lines
{K} => {O,E} : Confidence is 3/5 x 100% = 60%
{E} => {O,K} : Confidence is 3/4 x 100% = 75%
{K,E} => {O} : Confidence is 3/3 x 100% = 100%
{O,E} => {K} : Confidence is 3/3 x 100% = 100%
{O,K} => {E} : Confidence is 3/4 x 100% = 100%