Association Rules Report

Enzo Profli

1 Executive Summary

This report summarizes an effort to group stock keeping units (SKUs) into association rules, with the object of supplying a department store chain with item co-occurrence relationship data. The idea behind looking into these relationships is that the company can reshuffle items across its stores and put items that are frequently bought together in neighboring locations.

Our analysis has been focused on 10 stores that represent the store geographically across the United States - these stores were determined utilizing a K-medoid algorithm. Moreover, the analysis focuses on SKUs that generated significant revenue in these 10 stores. Given this dimensionality reduction, we were able to run the association rules algorithm to determine which SKU purchase relationships are strongest.

Using the lift measure (given sufficient support and confidence), we were able to determine the top 100 implication relationships in the transactions data. These implication relationships are the main product of this analysis, and they should guide the company when reshuffling 20 SKUs across its store - maximizing item purchases and revenue in the process. These full rundown on these rules can be seen in Appendix A.

2 Problem Statement

In this report, we look into a department store chain's transactions, in order to gather information on which items are frequently bought together. The objective is to supply the company management with relevant data to support future moves in stock keeping unit locations.

3 Assumptions

Some of the analysis is based on a few assumptions and shortcomings:

- The dataset is too large to conduct an analysis, and thus we must trim both the number of stores represented in the dataset and the number of SKUs available. The Methodology section explains how this was achieved.
- During store clustering, we assume that, if stores are geographically close, they are similar. This analysis does not account for other variables, such as city size, GDP per capita, etc. This is a shortcoming because it is possible that these stores do not accurately represent the median store customer. These chosen stores might, for example, be skewed towards small-town customers, which might have different tastes and preferences. The clustering procedure is further explained in the Methodology section.
- This analysis assumes that the variable group of STORE, TRANNUM, REGISTER and SALEDATE represent a unique transaction. Given this grouping, most of the transactions are one-SKU orders, which limits the power of our analysis.

4 Methodology

Given that the transactions dataset contains 120 million rows, we must subset this dataset in order to conduct this analysis, because of hardware limitations. In order to do so, we have conducted a K-medoid analysis to focus on 10 stores that represent the chain, at least geographically. By conducting this clustering, we get 10 "centralized" stores in which we will focus our efforts. Figure 1 displays the results of the clustering, as well as the stores that were picked for this analysis. We see that this Figure resembles that of the US map, and we can note that many stores are located in Florida, while there are far fewer stores in the pacific Northwest. This clustering procedure reduces our dataset to 3.7 million rows, representing a substantial improvement in performance.

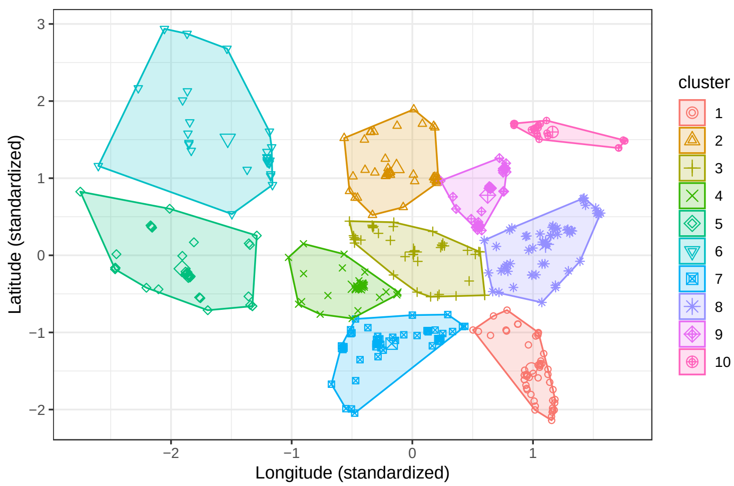

Figure 1: Store geographical clustering

Once we have this reduction in stores, we also look for reducing the number of SKUs in the dataset. To do so, we filter for SKUs that, over the dataset, represent revenues of $2,000 or more - this focuses or attention on products that bring in revenue for the company.

The next step in the process is defining transactions and baskets. Unfortunately, the transactions dataset does not possess an order ID in which we can identify transactions. So, we utilize the group of variables STORE, TRANNUM, REGISTER and SALEDATE to represent a transaction. This brings us a dataset containing around 450,000 transactions and 3500 SKUs - much more tractable for an association rules algorithm.

A few final details were set to run the association rules algorithm. We set the minimum support necessary for evaluation at 0.0001 (a given rule must occur at least .01% of the time) and minimum confidence at 0.1 (for item 1 → item 2, item 2 must have been bought in at least 10% of the orders containing item 1). Finally, we set the rule maximum length to 4 items, in order to keep the possible combinations at a reasonable amount, and the number of computations tractable for a common computer.

5 Analysis

Our association rules algorithm yielded 293 possible co-occurrence rules. Given that these rules already satisfy the support and confidence cutoffs, we will mostly focus on the lift measure to evaluate these rules, as it is the most important performance measure. Below, in Table 1 you can see the top 10 rules evaluated by the algorithm, by lift - in Appendix A you can check the top 100 rules.

In all of these cases, we see confidence above .5 and very high lift. Remember that, the farther away from 1 lift is, the better the association rule. We also see that these rules repeat themselves, with inverted order, which might indicate that these items are very frequently bought together, and not as standalone items. For example, if customers often bought item 2 when buying item 1, but customers often bought item 2 without item 1, only the rule {item 1} → {item 2} would appear. Finally, support is slightly above 0.0001 (0.01% of transactions) for each rule. This means they satisfy the constraint, but none of them are especially common.

Once we check all top 100 rules, support can be a bit larger - in some cases, support is above 0.0004 (0.04% of transactions).

Table 1: Top 10 Association Rules, by lift

Rule	support	confidence	lift
{7456422}=>{416422}	0.0001288	0.5490196	3060.362
{416422}=>{7456422}	0.0001288	0.7179487	3060.362
{376422}=>{7316422}	0.0001196	0.6265060	2929.017
{7316422}=>{376422}	0.0001196	0.5591398	2929.017
{1801637}=>{1761637}	0.0001288	0.6436782	2717.134
{1761637}=>{1801637}	0.0001288	0.5436893	2717.134
{1861637}=>{1821637}	0.0001334	0.6170213	2630.144
{1821637}=>{1861637}	0.0001334	0.5686275	2630.144
{9714273}=>{384274}	0.0001748	0.7102804	2174.808
{384274}=>{9714273}	0.0001748	0.5352113	2174.808

Unfortunately, the data on SKUs is not complete enough for us to understand what these items are, and if buying them together makes sense. In some cases, however, we do have some clues. For example, the second highest-lift pair (376422 $\rightarrow$ 7316422) apparently relates a set of bath towels (376422) to a set of hand towels (7316422) by the same brand (Crosscill). This is a pairing that is reasonable, and serves to validate our association rules analysis.

Given this analysis, we present 100 possible SKU moves that pairs items that are frequently bought together, potentially generating more revenue to the department store chain. With these 100 possibly promising moves, the company can choose 20 moves to be made across each store, while maximizing revenue.

6 Appendix A: Rules

Table 2: Top 100 Association Rules, by lift

Rule	support	confidence	lift
{7456422}=>{416422}	0.0001288	0.5490196	3060.3620
{416422}=>{7456422}	0.0001288	0.7179487	3060.3620
{376422}=>{7316422}	0.0001196	0.6265060	2929.0167
{7316422}=>{376422}	0.0001196	0.5591398	2929.0167
{1801637}=>{1761637}	0.0001288	0.6436782	2717.1342
{1761637}=>{1801637}	0.0001288	0.5436893	2717.1342
{1861637}=>{1821637}	0.0001334	0.6170213	2630.1439
{1821637}=>{1861637}	0.0001334	0.5686275	2630.1439
{9714273}=>{384274}	0.0001748	0.7102804	2174.8085
{384274}=>{9714273}	0.0001748	0.5352113	2174.8085
{4206421}=>{4456421}	0.0001702	0.7115385	2163.4252
{4456421}=>{4206421}	0.0001702	0.5174825	2163.4252
{6972521,7222521}=>{7232521}	0.0002093	0.8125000	1930.4201
{6972521,7232521}=>{7222521}	0.0002093	0.7222222	1869.1369
{768635}=>{748635}	0.0001610	0.5109489	1835.9957
{748635}=>{768635}	0.0001610	0.5785124	1835.9957
{7222521}=>{7232521}	0.0002829	0.7321429	1739.4994
{7232521}=>{7222521}	0.0002829	0.6721311	1739.4994
{4722472}=>{4752472}	0.0001955	0.6538462	1692.1772
{4752472}=>{4722472}	0.0001955	0.5059524	1692.1772
{4142521}=>{4462521}	0.0002645	0.7098765	1677.4306
{4462521}=>{4142521}	0.0002645	0.6250000	1677.4306
{5508634}=>{5548634}	0.0001932	0.6942149	1631.5551
{5548634}=>{5508634}	0.0001932	0.4540541	1631.5551
{8032644}=>{8042644}	0.0001242	0.4864865	1602.4201
{8042644}=>{8032644}	0.0001242	0.4090909	1602.4201
{7222521,7232521}=>{6972521}	0.0002093	0.7398374	1561.5238
{6372521}=>{6402521}	0.0002392	0.6265060	1521.7796
{6402521}=>{6372521}	0.0002392	0.5810056	1521.7796
{8530723}=>{8520723}	0.0001932	0.6268657	1465.3491
{8520723}=>{8530723}	0.0001932	0.4516129	1465.3491
{738635}=>{768635}	0.0001035	0.4591837	1457.2881
{768635}=>{738635}	0.0001035	0.3284672	1457.2881
{7232521}=>{6972521}	0.0002898	0.6885246	1453.2214
{6972521}=>{7232521}	0.0002898	0.6116505	1453.2214
{7222521}=>{6972521}	0.0002576	0.6666667	1407.0874
{6972521}=>{7222521}	0.0002576	0.5436893	1407.0874
{4662472}=>{4752472}	0.0001380	0.5309735	1374.1783
{4752472}=>{4662472}	0.0001380	0.3571429	1374.1783
{8412644}=>{8402644}	0.0002438	0.5988701	1295.4364
{8402644}=>{8412644}	0.0002438	0.5273632	1295.4364
{6042521,6062521}=>{6072521}	0.0001081	0.8245614	1249.1674
{6642521}=>{6742521}	0.0003266	0.6794258	1241.2082
{6742521}=>{6642521}	0.0003266	0.5966387	1241.2082
{6032521,6072521}=>{6062521}	0.0003013	0.8187500	1210.8310
{6032521,6062521}=>{6072521}	0.0003013	0.7891566	1195.5310
{6042521,6072521}=>{6062521}	0.0001081	0.7966102	1178.0889

{8132644}=>{8122644}	0.0002323	0.5179487	1166.8338
{8122644}=>{8132644}	0.0002323	0.5233161	1166.8338
{6062521,6072521}=>{6032521}	0.0003013	0.6550000	1152.9856
{6062521,6072521}=>{6042521}	0.0001081	0.2350000	1032.0773
{6072521}=>{6062521}	0.0004600	0.6968641	1030.5767
{6062521}=>{6072521}	0.0004600	0.6802721	1030.5767
{6032521}=>{6062521}	0.0003818	0.6720648	993.9015
{6062521}=>{6032521}	0.0003818	0.5646259	993.9015
{6032521}=>{6072521}	0.0003680	0.6477733	981.3427
{6072521}=>{6032521}	0.0003680	0.5574913	981.3427
{576156}=>{2682771}	0.0001012	0.3437500	970.5134
{2682771}=>{576156}	0.0001012	0.2857143	970.5134
{6042521}=>{6072521}	0.0001357	0.5959596	902.8476
{6072521}=>{6042521}	0.0001357	0.2055749	902.8476
{6042521}=>{6062521}	0.0001311	0.5757576	851.4750
{6062521}=>{6042521}	0.0001311	0.1938776	851.4750
{5453386,7248011}=>{7218011}	0.0001035	0.5625000	764.2793
{4702798}=>{3782798}	0.0001334	0.3240223	749.3706
{3782798}=>{4702798}	0.0001334	0.3085106	749.3706
{8976664}=>{6876664}	0.0001081	0.2716763	674.9836
{6876664}=>{8976664}	0.0001081	0.2685714	674.9836
{5453386,7218011}=>{7248011}	0.0001035	0.5172414	673.3275
{4562798}=>{3742798}	0.0001196	0.2694301	650.8083
{3742798}=>{4562798}	0.0001196	0.2888889	650.8083
{2703090}=>{2893090}	0.0003243	0.4930070	630.4544
{2893090}=>{2703090}	0.0003243	0.4147059	630.4544
{4472217}=>{7351914}	0.0003059	0.8926174	584.4897
{7351914}=>{4472217}	0.0003059	0.2003012	584.4897
{6939904}=>{5129905}	0.0002599	0.3704918	561.2757
{5129905}=>{6939904}	0.0002599	0.3937282	561.2757
{9628964}=>{2168966}	0.0001265	0.3160920	509.0134
{2168966}=>{9628964}	0.0001265	0.2037037	509.0134
{3772798}=>{4572798}	0.0001219	0.2345133	443.3219
{4572798}=>{3772798}	0.0001219	0.2304348	443.3219
{2168966}=>{9358964}	0.0001472	0.2370370	434.8579
{9358964}=>{2168966}	0.0001472	0.2700422	434.8579
{6349904}=>{5369905}	0.0003496	0.3743842	432.9216
{5369905}=>{6349904}	0.0003496	0.4042553	432.9216
{2494717}=>{144717}	0.0001058	0.2527473	422.6615
{144717}=>{2494717}	0.0001058	0.1769231	422.6615
{5309905}=>{7029904}	0.0004531	0.4624413	421.5196
{7029904}=>{5309905}	0.0004531	0.4129979	421.5196
{8888965}=>{9358964}	0.0001012	0.2189055	401.5946
{9358964}=>{8888965}	0.0001012	0.1856540	401.5946
{8888965}=>{2168966}	0.0001150	0.2487562	400.5804
{2168966}=>{8888965}	0.0001150	0.1851852	400.5804
{5749904}=>{5109905}	0.0004209	0.3828452	396.3268
{5109905}=>{5749904}	0.0004209	0.4357143	396.3268
{7248011}=>{7218011}	0.0002116	0.2754491	374.2579
{7218011}=>{7248011}	0.0002116	0.2875000	374.2579
{7258011}=>{7228011}	0.0001633	0.2034384	340.2038

{7228011}=>{7258011}	0.0001633	0.2730769	340.2038
{5189905}=>{6949904}	0.0004508	0.4355556	333.9951
{6949904}=>{5189905}	0.0004508	0.3456790	333.9951