Project 1 – Market Basket Analysis

*Joseph Park - x172917 Date: 4/12/20*

Objective

The assignment is an illustration of unsupervised learning (more specifically market basket analysis using the Apriori algorithm) based on association rules using particle collision data from a particle accelerator to identify said rules between combinations of particles and quantiles of their features.

Market basket analysis with Apriori algorithm

Step 1: Download the data set from Kaggle, look for missing data and look at summary, size and shape.

Step 2: Since the data set is too large to process, randomly sample.

Step 3: Since the particle data is not the sort of data this was intended for (a list of combinations of repeated items), we need to do some reformatting given in the next few following steps.

Step 4: Partition each feature (excluding id) into some number of bins and some size each. Though the Apriori algorithm is robust to varying sizes of bins and/or numbers of data points per bin, I figured I might as well make things simpler and work with quantiles.

Step 5: We start by copying the 50,000 sample to keep the id feature and replace each of the other elements with a label of the bin we assign them to.

Step 6: I initially tried this brute force with cases (per quantile) nested in a loop (iterating over the rows/particles) nested in a loop (iterating over the columns/features). The time complexity was way too much.

Step 7: Next, I tried using quintiles with the bin function (looped over the features), but the breaks were not unique for the fifth feature (as demonstrated by the histogram below).

Step 8: I then use the cut function with cases (as you can see in the knitted html) with in a loop iterating feature index, where for the breaks, I use the quantile function (this time doing quartiles) plus a very small amount that barely increases per each break.

Step 9: Per each case (quartile), we assign to each element a value 1 through 4 corresponding to the quartile of its feature.

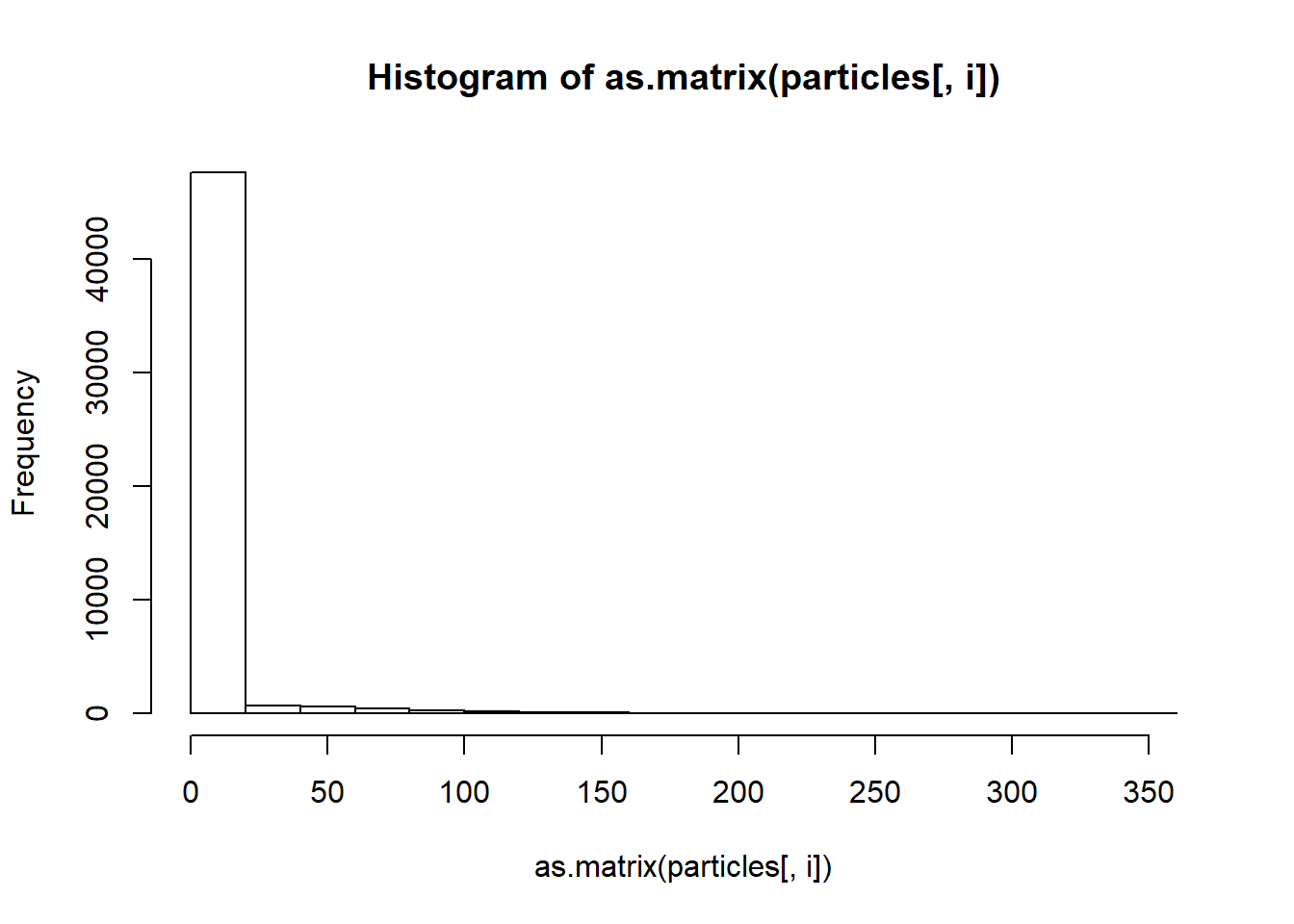
Step 10: Iterating over the features (excluding id), we add 10 times the index of the feature to each of its element. This way each “item” (representing a different feature and quartile) is unique and easily identifiable by increasing order in both directions (dimensions) of the indices.

Step 11: Write this dataframe to a csv and then plug the csv into the read.transaction function to reformat the data (into binary) for the Apriori algorithm.

Step 12: Use the itemFrequency function to discover that over 90% of the data is in the first supposed quartile while there is none in the second or third of the fifth feature (as also demonstrated by the histogram below). There are similar issues in other features. There are just that many zeros. Luckily, as mentioned above, Apriori is robust enough to handle this. (One possibility to get better results is replacing each zero with NaN.)

Step 13: Run the Apiori algorithm on the transformed (binary) data.

Step 14: Look through the associations for insightful rules (potentially useful for classification trees and rule, e.g. the rule {21, 43} => {211} which has a confidence of 0.997, support of 0.114, and lift of 1.78, or the 3 rules that I have listed in the results with only one element on the left hand side and only a particle label on the right.



The above histogram of the fifth feature that corresponds to Step 7 and Step 12.

Results

## Formal class 'transactions' [package "arules"] with 3 slots

## ..@ data :Formal class 'ngCMatrix' [package "Matrix"] with 5 slots

## .. .. ..@ i : int [1:350000] 2 3 11 14 16 18 22 4 6 7 ...

## .. .. ..@ p : int [1:50001] 0 7 14 21 28 35 42 49 56 63 ...

## .. .. ..@ Dim : int [1:2] 26 50000

## .. .. ..@ Dimnames:List of 2

## .. .. .. ..$ : NULL

## .. .. .. ..$ : NULL

## .. .. ..@ factors : list()

## ..@ itemInfo :'data.frame': 26 obs. of 1 variable:

## .. ..$ labels: chr [1:26] "-11" "21" "211" "22" ...

## ..@ itemsetInfo:'data.frame': 0 obs. of 0 variables

## NULL

^ Graphic 1 is the structure of the transaction file.

## transactions as itemMatrix in sparse format with

## 50000 rows (elements/itemsets/transactions) and

## 26 columns (items) and a density of 0.2692308

##

## most frequent items:

## 51 211 71 2212 61 (Other)

## 46730 27945 21747 19588 17138 216852

##

## element (itemset/transaction) length distribution:

## sizes

## 7

## 50000

##

## Min. 1st Qu. Median Mean 3rd Qu. Max.

## 7 7 7 7 7 7

##

## includes extended item information - examples:

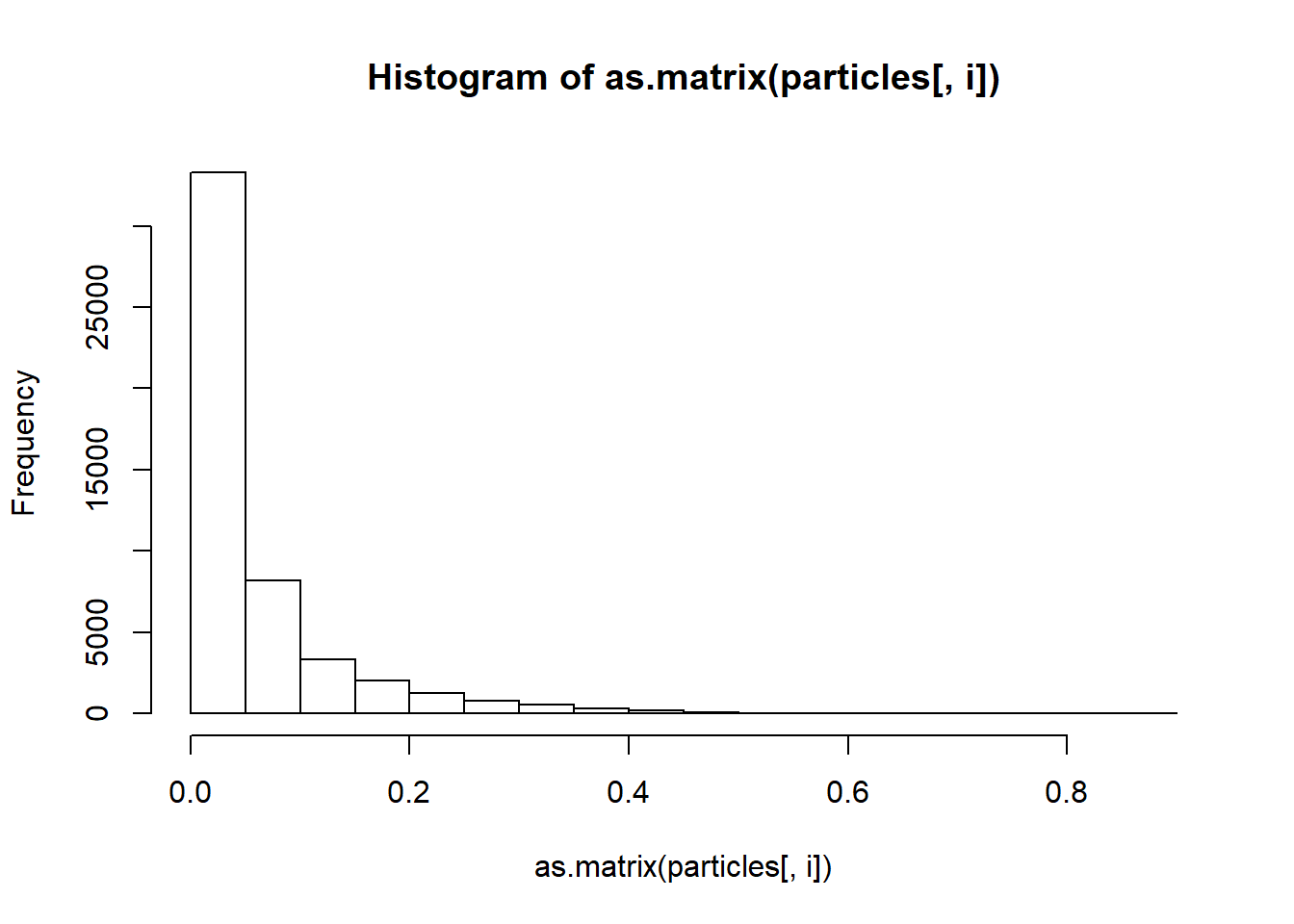
## labels

## 1 -11

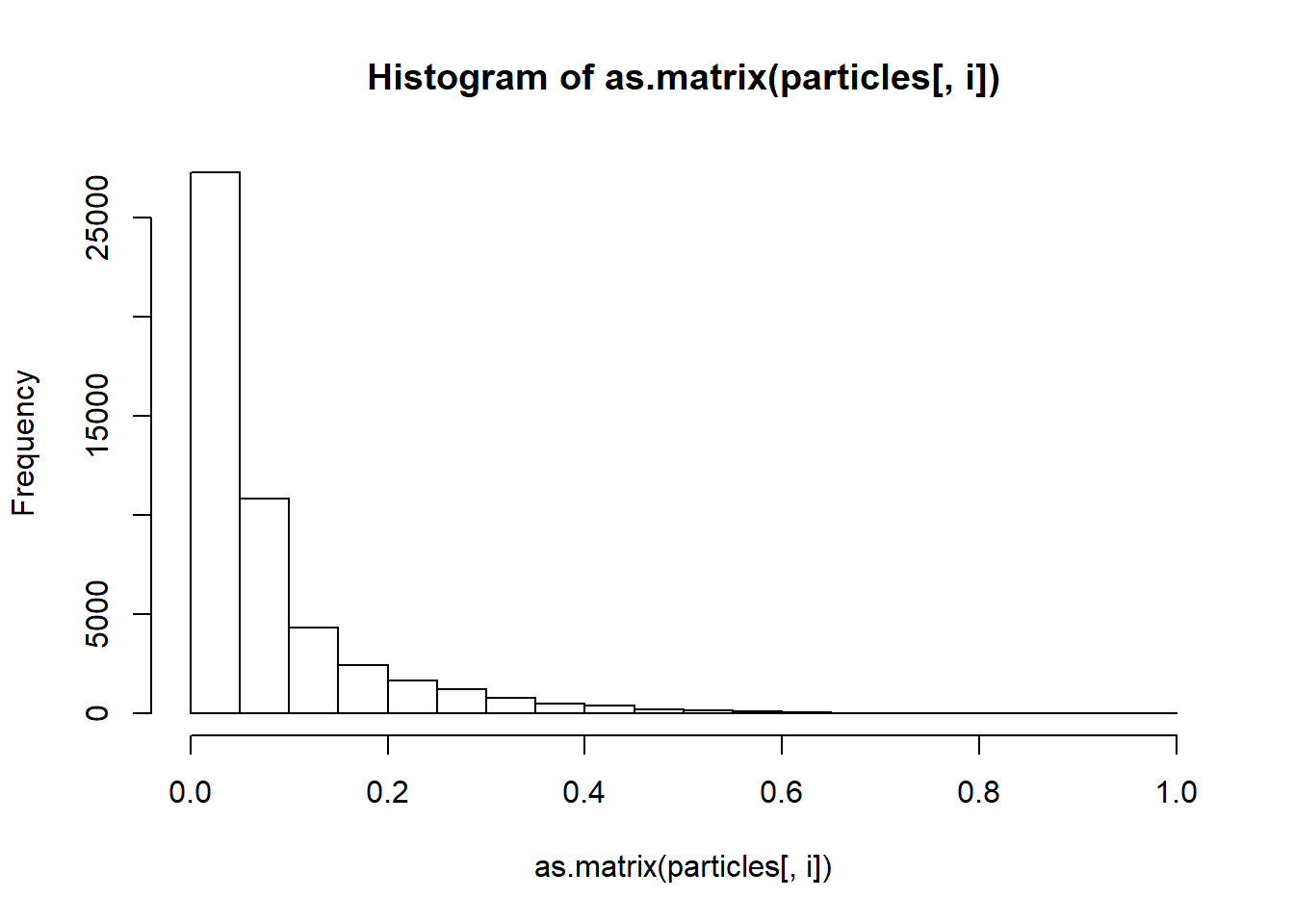
## 2 21

## 3 211

^ Graphic 2 is the summary of the transactions file.



^ Graphic 3 is the histogram of the sixth feature



^ Graphic 4 is the histogram of the seventh feature

-11 21 211 22 2212 23 24 31 32 321

0.00318 0.25010 0.55890 0.25000 0.39176 0.24990 0.25000 0.25012 0.25000 0.046

33 34 41 42 43 44 51 54 61 62

0.24992 0.24996 0.25006 0.25018 0.25064 0.24912 0.93460 0.06540 0.34276 0.158

63 64 71 72 73 74

0.24892 0.24966 0.43494 0.06588 0.24922 0.24996

^ Graphic 5 is the item frequency of the transaction file.

## lhs rhs support confidence lift count

## [1] {-11,73} => {54} 0.00022 1.0000000 15.290520 11

## [2] {-11,72} => {44} 0.00030 1.0000000 4.014130 15

## [3] {-11,73} => {44} 0.00022 1.0000000 4.014130 11

## [4] {-11,23} => {44} 0.00020 1.0000000 4.014130 10

## [5] {-11,73} => {64} 0.00022 1.0000000 4.005447 11

## [6] {-11,24} => {64} 0.00016 1.0000000 4.005447 8

## [7] {-11,24} => {74} 0.00016 1.0000000 4.000640 8

## [8] {-11,61} => {71} 0.00182 1.0000000 2.299168 91

## [9] {34,54} => {71} 0.02042 0.9903007 2.276867 1021

## [10] {23,44} => {211} 0.07278 0.9961675 1.782372 3639

^ Graphic 6 lists the first ten rules apparently ordered by confidence.

## lhs rhs support confidence lift count

## [1] {22,42,54,63} => {321} 0.00020 1 21.66378 10

## [2] {22,42,62,72} => {321} 0.00014 1 21.66378 7

## [3] {22,32,42,72} => {321} 0.00014 1 21.66378 7

## [4] {22,42,62,74} => {321} 0.00020 1 21.66378 10

## [5] {21,41,64,73} => {321} 0.00010 1 21.66378 5

## [6] {22,42,61,73} => {321} 0.00016 1 21.66378 8

## [7] {22,32,42,74} => {321} 0.00022 1 21.66378 11

## [8] {22,42,54,63,73} => {321} 0.00012 1 21.66378 6

## [9] {22,33,42,54,63} => {321} 0.00010 1 21.66378 5

## [10] {22,42,51,62,72} => {321} 0.00014 1 21.66378 7

^ Graphic 7 lists the first ten rules with particle 321 as the consequent (i.e. in the RHS).

## lhs rhs support confidence lift count

## [1] {23,44} => {211} 0.07278 0.9961675 1.782372 3639

## [2] {22,44} => {211} 0.05952 0.9913391 1.773733 2976

## [3] {34,43} => {211} 0.09036 0.9962514 1.782522 4518

## [4] {22,43} => {211} 0.05706 0.9989496 1.787349 2853

## [5] {21,42} => {211} 0.10430 0.9994251 1.788200 5215

## [6] {21,43} => {211} 0.11360 0.9966661 1.783264 5680

## [7] {44,54,63} => {211} 0.00424 0.9906542 1.772507 212

## [8] {33,43,54} => {211} 0.00414 0.9951923 1.780627 207

## [9] {34,42,54} => {211} 0.00974 0.9979508 1.785562 487

## [10] {34,43,54} => {211} 0.00796 0.9950000 1.780283 398

^ Graphic 8 lists the first ten rules with particle 211 as the consequent.

## lhs rhs support confidence lift count

## [1] {44} => {211} 0.23800 0.9553629 1.709363 11900

## [2] {41} => {2212} 0.23648 0.9456930 2.413960 11824

## [3] {21} => {211} 0.23418 0.9363455 1.675336 11709

^ Graphic 9 lists the top three rules wherein there is one antecedent and only a particle as a consequent.

Interpretation of the Results

Note 1: If particle -11 was of particular interest, a supercomputer (and thus ability to process a sample size significantly more 50,000) would be of use, as 159 (especially out of 50,000) did not give very useful results.

Note 2: Graphic 1 says there are 26 observations. This is explained by 4 labels per each of the 7 features minus the labels of the quartiles that contained no data due to all the zeros.

Note 3: It also has 7 listed for a host of different metrics on transaction size. This makes since because we built every one of the “transactions” as a particle with 6 feature quantiles.

Note 4: The items listed as repeating the most in Graphic 2 are the particles that are far more prevalent than the other two and the first quartile of features that have the most zeros. You can see this in Graphic 3, Graphic 4 and the histogram above “Results.”

Note 5: The labels of quantiles in Graphic 5 that are not quartiles can be explained again by all the zeros of the fifth, sixth, and seventh features.

Note 6: Particle -11 being listed in the first 8 rules in Graphic 6 makes sense as there are so few particle -11 entries which is why some of the rules it is in have a confidence of 100%. (This is a matter of coarse resolution. More data would give finer resolution and start knocking down some of those 100%’s.)

Note 7: The two less prevalent particles have weak results in terms of what combinations imply them as evidenced by implications (rules) listed for 321 in Graphic 7 and the lack of any for particle -11 (as you can see in the knitted html). The rules start with 4 antecedents for particle 321 as shown in Graphic 7 and 2 for particle 211 as shown in Graphic 8.

Note 8: As noted in Step 14 and seen in Graphic 8, potentially useful for classification trees or rules is the rule {21, 43} => {211} which has a confidence of 0.997, support of 0.114, and lift of 1.78. What makes it stronger, as seen in the knitted html (via lack of an output from the corresponding inspect function), is that it is the only one with a rule with a particle as the consequent and {21, 43} as the antecedent.

Note 9: Also noted in Step 14 and seen in Graphic 9, potentially useful for classification trees or rules are the 3 rules that have only one element on the left hand side and only a particle label on the right, in particular as the first node or rule (since there is only one antecedent and they are the top 3). (In the label of the graph, it did not need to say the antecedents were not particles because particles are mutually exclusive in these rule as they are in the same feature, which is of course true for the quantiles in the same feature.)