COSC 757 Data Mining Assignment 4

Mary Snyder

Department of Computer & Information Sciences  
College of Science & Mathematics  
Towson University

msnyde8@students.towson.edu

**ABSTRACT**

In this paper, I will be exploring a dataset to become more familiar with frequent itemset and association rule mining through the COSC 757 Data Mining Assignment 4.

**Categories and Subject Descriptors**

H.2.8 **[Database Management]** Database Applications – *Data mining*

**Keywords**

Frequent Itemset; Association Rule Mining; Multivariate; Categorical; arules; Apriori Algorithm; Eclat Algorithm; Support; Confidence;

# INTRODUCTION

## Dataset

I chose a dataset from the UCI Machine Learning Repository whose attributes are categorical in nature. This Congressional Voting Records dataset comprises United States House of Representatives Congressmen voting records for the 16 key votes for 98th Congress second session in 1984 as identified by the Congressional Quarterly Almanac (CQA). The dataset contains information regarding the voter’s party (Democrat or Republican) as well as a simplified vote of yea/nay/abstain for each of the key votes. There were originally 9 different kinds of votes which were simplified into the previously mentioned yea/nay/abstain as follows: voted for, paired for, and announced for are marked as a yea,; voted against, paired against, and announced against are marked as a nay; voted present, voted present to avoid conflict of interest, and did not vote or otherwise make a position known are marked as abstain. There are 435 instances with no missing values. Key votes of abstain are indicated by a ‘?’ value. There are 17 attributes: Class Name (Democrat, Republican) and a Boolean value for each of the 16 key votes.

## Objective of Analysis

The objective of frequent pattern analysis is to find inherent regularizes in the data. A frequent patter reveals an intrinsic and important property of the dataset and mining of these patterns is the foundation for many essential data mining tasks including association, correlation, and causality analysis.

Frequent pattern analysis is achieved through use of itemsets. An itemset is a set of one or more items. The relative support of an itemset is the fraction of transactions that contain the itemset out of the total number of itemsets. An itemset is considered frequent if the support for the itemset is no less than a specified threshold.

# METHODOLOGY

## Preprocessing

The dataset contains voting records for 16 key votes; however, each vote was recorded in the dataset as a simple y for yea, n for nay, or ? for abstain (see Table 1). Since each representative (defined by a row) could have voted yea for multiple key votes, this caused issues when trying to treat the data as transactions in order to basket the data for analysis. To overcome this issue I updated the values for each of the key votes to indicate not only the vote (yea/nay/abstain), but also the vote number (see Table 2). For example, a ‘y’ or yea vote for the first key vote (column V2) would translate into ‘1y’ for the processed data, a ‘n’ or nay vote for the fifth key vote (column V5) would translate into ‘5n’, and a ‘?’ or abstain vote for the sixteenth key vote would translate into ‘16?’ for the processed data.

Table 1. Sample of Original Data

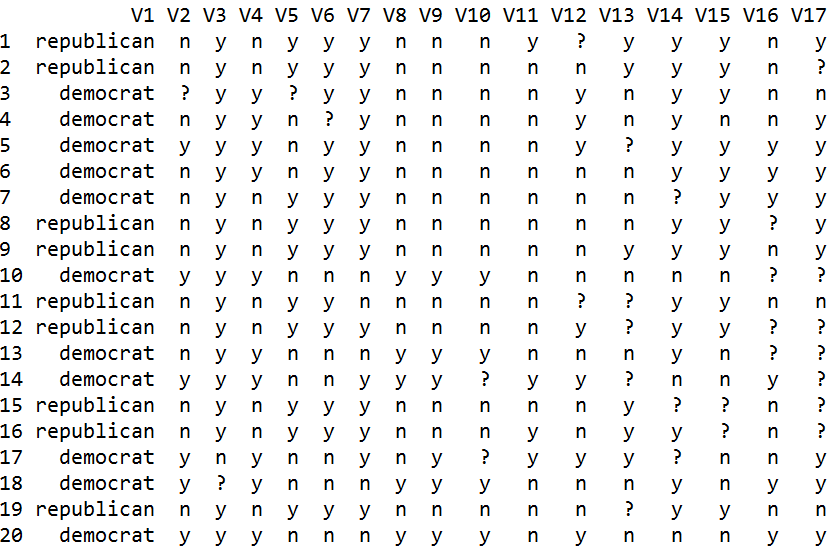
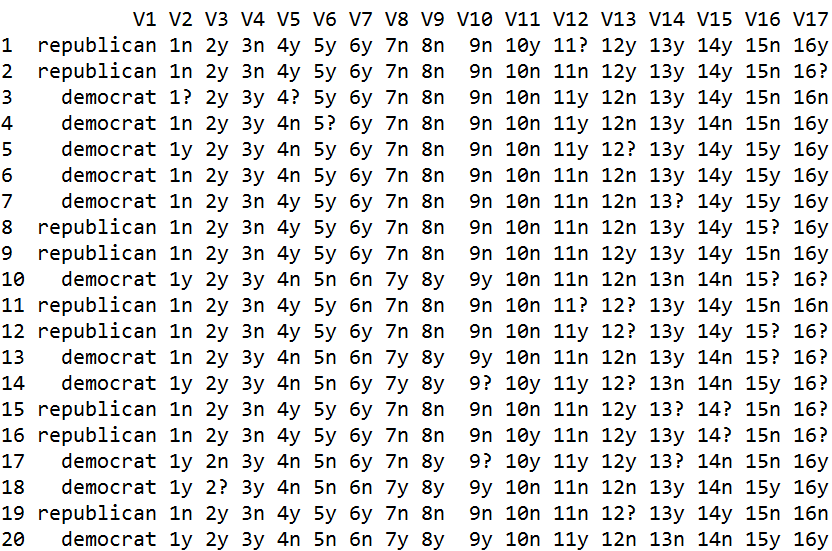


Table . Sample of Processed Data



There are two keyword attributes, which varied in their degree of granularity, one more simplistic (Keywords) and one more categorical (High-Level Keyword(s)). I chose to eliminate the more simplistic Keywords attribute in favor of the more categorical High-Level Keyword(s) in hope this would produce better clustering results. The remaining attribute Topics also seemed fairly categorized, so I chose to pair it with the High-Level Keyword(s) for the analysis.

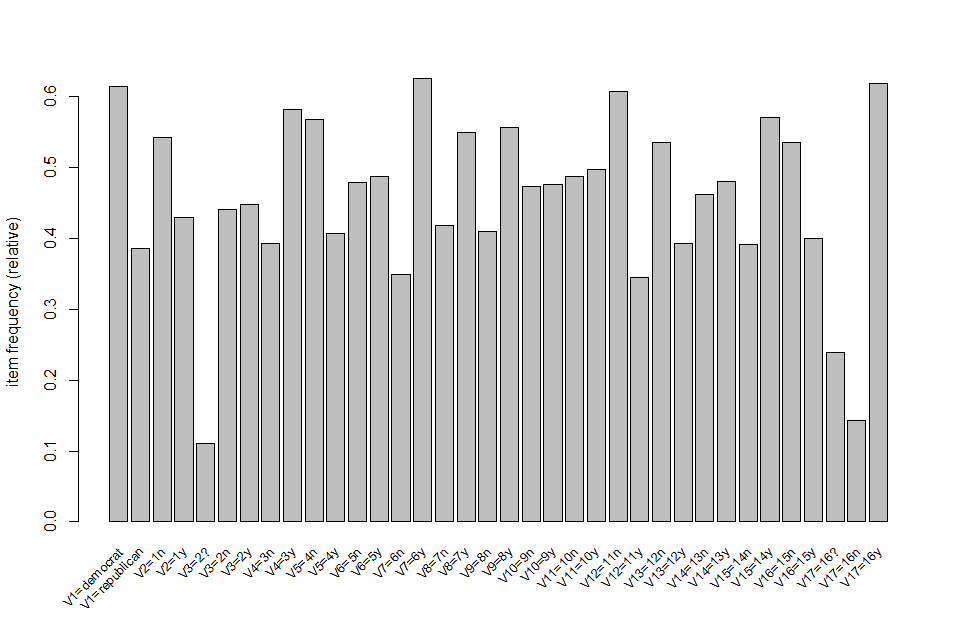


Figure . Item Frequency

## Tools/Approaches

The statistical computing environment and language R contains a package ‘arules’ to assist in Frequent itemset or association rule mining. There are two specific tools in the arules package that will be used for this analysis: Apriori and Eclat.

### Apriori

Apriori in the R arules package is used to mine frequent datasets, association rules, or association hyperedges using the Apriori algorithm. The Apriori algorithm conducts level-wise searches for frequent itemsets as follows:

Ck: Candidate itemset of size k

Lk : frequent itemset of size k

L1 = {frequent items};

for (k = 1; Lk != Ø; k++) do begin

Ck+1 = candidates generated from Lk;

for each transaction *t* in database do

increment the count of all candidates in Ck+1 that are contained in *t*

Lk+1 = candidates in Ck+1 with min\_support

end

return Uk Lk;

There are a few disadvantages of the Apriori algorithm. It requires multiple scans of the database, specifically *n* + 1, where *n* is the length of the longest pattern. In addition, the candidate generation can result in very large candidate sets. For example, if the dataset contains 104 frequent 1-itemsets, the algorithm will need to generate 107 candidate 2-itemsets.

### Eclat

Eclat in the R arules package is used to mine frequent datasets using the Eclat algorithm. The Eclat algorithm uses bottom-up lattice traversal and simple intersection operations for equivalence class clustering. The algorithm is defined recursively with the initial call using all the single items with their transaction ids (tids) and each recursive call examines the intersections of pairs of tids to generate new candidates. Determine support of any k-itemset by intersecting tid-lists of two of its (k-1) subsets.

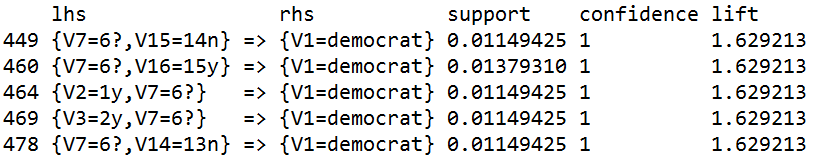
While the Eclat algorithm is very fast in support counting, it is not without its weaknesses. The intermediate tid-lists may become too large for memory. When the tid-list is large, at the time of computation it requires a large amount of space to store the candidate sets. In addition, the algorithm takes more time for computing intersections when the tid-list is large.

# RESULTS

## Apriori

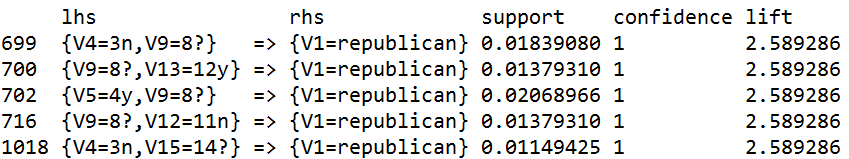
For both partitioning methods, a *k* value for the number of partitions needs to be specified ahead of time. To determine the best value I used the Elbow method, which looks at the sum of squared error (SSE) within groups as a function of the number of clusters (see Figure 3). Looking for the bend of elbow in the plot gives a good indication of a value for *k*. In this case, 8 or 10 clusters seemed to be good bend/elbow locations so I used both those values for *k* in my analysis.

Table . Apriori Democrat Confidence Sort



Running k-means clustering on the data for *k*=10 produced less great of results than with *k*=8 (see Figure 3) with a few possible visible outliers.

Table . Apriori Republican Confidence Sort



## Eclat

I first ran the DBSCAN with an Eps value of 5. The results (see Figure 10. DBSCAN Clustering for Eps=5Figure 10) showed some clustering but still many outliers. I then re-ran the DBSCAN with an Eps value of 10. The results (see Figure 11) were much improved; however, there were still a few outliers. I decided to see if it was possible to eliminate the remaining outliers. To do this I again increased the Eps value to 15 and re-ran the test. This result (see Figure 12) was not well clustered at all, so I knew I needed a Eps value closer to the last good clustering or Eps of 10. I tried a few other Eps values (see results for Eps 11 in Figure 13 and Eps 11.5 in Figure 14) until I found the results I considered the best for Eps value of 11.1 (see Figure 15) with as few outliers as possible.

Table . Eclat Democrat Support Sort

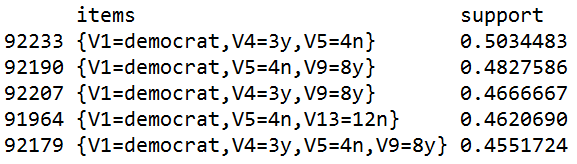
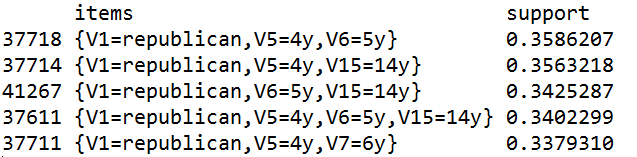


Table . Eclat Republican Support Sort



# CONCLUSIONS

## Evaluation Metrics

### Support

The support value of an itemset is defined as the proportion of transactions in the database that contain the itemset. It can be written as supp(*X*), where *X* is an itemset.

### Confidence

The confidence value of a rule is the proportion of the transactions that contain an itemset, that also contain another non-intersecting itemset of the same dataset. It can be written as conf(*X* => *Y*) = supp(*X* U *Y*) / supp(*X*), where *X* and Y are itemsets such that *X* ∩ *Y* = Ø and *X* => *Y* is a rule.

## Apriori

Using *k*=8 for the k-medoids partitioning approach gave the best average silhouette width for the k-medoids method, but still slightly lower than the k-means for k=8. The value for the silhouette width was 0.52 and the silhouette plot shows outliers (see Figure 19).

Using *k*=10 for the k-medoids partitioning approach gave a slightly lower value for the silhouette width than with *k*=8, but still not bad at 0.49 (see Figure 20). The silhouette plot shows an outlier, but still seems to fit the data fairly well.

Table . Apriori Democrat Support Sort Filter

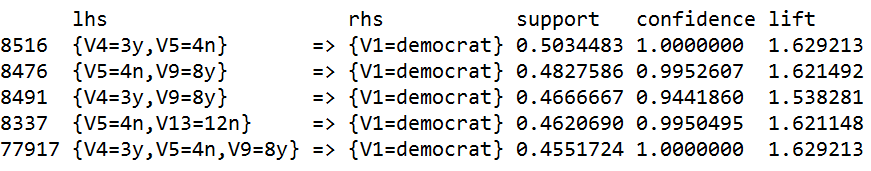
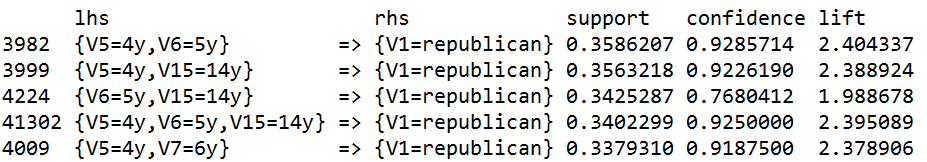


Table . Apriori Republican Support Sort Filter



## Eclat

The silhouette plot for DBSCAN with Eps of 11.1 shows even with this value there were still a few outliers. In addition, the average silhouette width is lower than either of the *k* values used in both the k-means and the k-medoids approaches. So for this dataset, the density-based approach did not do as well as the either of the partitioning approaches. This may be the case because the clusters determined by the partitioning approaches are larger to encompass all the data, but less not very dense especially at the edges. This would lead to the nodes near the edges not being included by DBSCAN and therefore there would be more outliers in the resulting DBSCAN clustering.

Table . Eclat Frequent Itemsets – Top 5

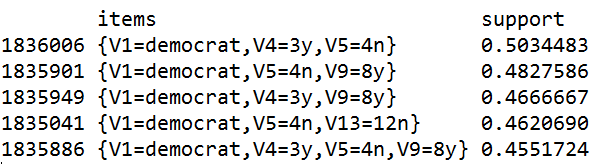
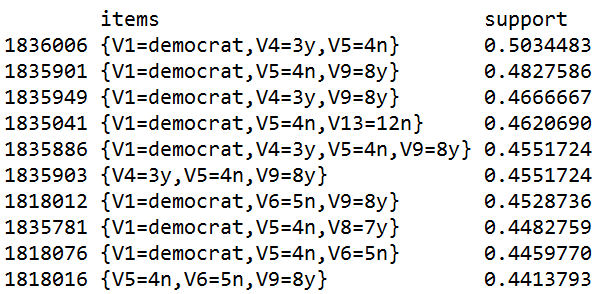


Table . Eclat Frequent Itemsets – Top 10



# REFERENCES

1. Larose, D. and Larose, C.D. *Discovering Knowledge in Data: An Introduction to Data Mining*. Wiley-Interscience, 2014.
2. Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
3. Congressional Quarterly Almanac, 98th Congress, 2nd session 1984, Volume XL: Congressional Quarterly Inc. Washington, D.C., 1985. (Shlimmer, 1987)
4. Schlimmer, J.C. (1987). Concept acquisition through representational adjustment. Doctoral dissertation, Department of Information and Computer Science, University of California, Irvine, CA.