**Association Rule and Frequent Itemset Mining**

**Spring 2016**

The source for this exercise is the documentation for the arules R package (Apriori example) and the “Data Mining Algorithms in R Wikibook” (Eclat example) available respectively at the following websites:

<http://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R>

<http://rss.acs.unt.edu/Rdoc/library/arules/doc/arules.pdf>

Please keep in mind, that while this exercise uses R, there are other software that can be used to mine frequent itemsets such as Weka, Rapid Miner, KNIME, and the freely available code on Christian Borgelt’s website (<http://www.borgelt.net//fpm.html>). Feel free to explore these tools as you complete Assignment 4.

The arules package in R contains methods for frequent itemset mining including both the Apriori and Eclat algorithms. In this lab, we will demonstrate both of these algorithms using the Adult dataset which is data from the U.S. Census Bureau containing 48842 instances with 14 attributes such as age, work class, education, etc…

**Association Rule Mining with R**

Open a new R session

Go to the Packages menu item and select Install Package(s)

After selecting a mirror select the arules package to be installed.

Once the arules package is installed, go to the Packages menu item and select Load Package and select the arules package

**Explore the data**

The Adult dataset comes with the arules package and has been preprocessed as a binary incidence matrix with 48824 transactions (rows) and 115 items.

Load the Adult dataset

> library(datasets)

> data(Adult)

Inspect the dataset

> summary(Adult)

> Adult[1:2,]

The summary of the transaction gives an overview of the data showing the most frequent items, the length distribution of the transactions and the extended item information which shows which variable and which value were used to create each binary item.

We can now use itemFrequencyPlot() to show which items are important in the dataset. Here we will create an itemFrequencyPlot with a support greater than 10%.

> itemFrequencyPlot(Adult, support=0.1, cex.names = 0.8)

Explore the plot. Which items are the most “important” in terms of having a high frequency?

**Apriori Algorithm**

Now that we have a feel for the data, we can find association rules in the dataset using the apriori() function. For this example, we will find rules with a minimum support of 1% and a confidence of 0.6.

> rules <- apriori(Adult, parameter = list(support = 0.01, confidence = 0.6))

How many rules were generated?

As we discussed in the lecture, association rule mining creates a large number of rules. To analyze the rules, subset() can be used to produce separate subsets of rules. For example, if we want to look at the rules that have the variable income on the right-hand-side of the rule we would use the following command:

> rulesIncomeSmall <- subset(rules, subset = rhs %in% “income=small” & lift > 1.2)

> rulesIncomeLarge <- subset(rules, subset = rhs %in% “income=large” & lift > 1.2)

Notice that we can filter the rules using the lift measure. In this example, we require the lift to exceed 1.2. We can now explore the generated rules using the inspec() and sort() functions. For example, if we want to explore rulesIncomeSmall sorted by confidence, we can issue the following command to find the top 3 rules:

> inspect(sort(rulesIncomeSmall, by = "confidence")[1:3])

What rules did this command find? Now feel free to explore the rule set based on a number of attributes to see what associations are present in the dataset.

**Eclat Algorithm**

We can also demonstrate the Eclat algorithm in finding frequent itemsets in the Adult dataset. For example, if we want to find frequent itemsets with a minimum support of 10% and a max length of 15 items, we would issue the following command:

itemsets <- eclat(Adult, parameter = list(sup = 0.1, maxlen = 15))

How many itemsets were generated?

Now we want to refine our search a bit and find frequent itemsets with a minimum support of 50%

> fsets <- eclat(Adult, parameter = list(sup = 0.5))

Now we can display the top 5 itemsets with the highest support

> fsets.top5 <- sort(fsets)[1:5]

Explore the dataset using the Eclat algorithm and see if you can find the frequent itemsets that generated the same rules found in the Apriori example