**Final Report**

by

Team “Wei

Too Good”

**Team Members:** James Keats and David Horntvedt

12/11/2015

Table of Contents

1. Introduction

2. Background

3. Implementation Detail

4. Experimentation Detail

5. Discussion and Conclusion

6. Reference

7. Appendix

# **1.** **Introduction**

Data mining is an important topic in today’s data-driven society. The prevalence of computing is making data easier and easier to calculate and manipulate. Association mining specifically is about finding relationships between different sets of data. For example, looking at all of the transactions of a retail store to find out which items were commonly bought together, so that you can place them near each other inside of the store.

# **2.** **Background**

Apriori is an algorithm used for association mining. Apriori works by only considering combinations of itemsets that have previously been found to be large, or to meet the minimum support that the user specified. For every k-itemset, Apriori generates a list of candidates from the k-1itemsets that were previously found to be large. It then passes over the transaction database to determine if each candidate in the k-itemset is large. If it can generate more candidates, it will then increment k and continue to loop until no more candidates are generated and found large.

One of the main advantages of Apriori is that at every step along the way, it only considers candidates which have the potential to be large. This saves a lot of time by not even considering the vast majority of potential candidates and combinations. There are also disadvantages, however. The primary among these is that it can take many different passes over the data to determine which candidates meet minimum support, and so an efficient way to check support is necessary. In addition, it is not possible, at least in the most basic version of Apriori, to limit your results to certain itemsets.

# **3.** **Implementation Detail**

We used the following data structures and algorithms in our implementation:

* Apriori Result: A struct created specifically for running the different tests of the apriori algorithm, and holding the data used in the charts below. This struct contained an integer representing what k value we were at, an integer representing the number of itemsets found at this k value, and a double representing the number of seconds required to calculate this set of k-itemsets.
* Dynamic Array: A template class that served exactly what it sounds like. Somewhat similar to std::vector in that it grows and shrinks as needed by the number of items currently stored in it. Contained a pointer to an array of type T, as well as an integer representing the actual capacity of the array in memory, separate from the number of indexes filled, or the “count” of items in the array. In our implementation, Dynamic Array was used to hold integers to represent the itemsets, as well as an Itemset struct, other Dynamic Arrays, and more.  
  The main advantage of this structure is its ability to hold a variety of different data types and sizes. The flexibility of a structure with excellent random access that grows and shrinks as needed cannot be denied. The main disadvantage, however, is the extra time needed to grow and shrink the array as this involves copying large amounts of data. In addition, the structure can use more memory than is necessary depending on the method used to grow and shrink it. In our case, for simplicity’s sake, we used a linear growth rate which resulted in an overly large array being allocated for small itemsets, and not enough being allocated for large ones (resulting in expandArray() being called more times than we would’ve liked). In retrospect, I would have preferred to use exponential growth similar to std::vector in order to save more time and ensure an amortized constant time for insertion.
* Itemset: A struct used to hold an itemset. Contained a Dynamic Array of integers (the itemset itself), and a boolean representing whether that set met minimum support. This existed simply to keep the code cleaner, and group related variables together logically.
* Itemset Holder: This class was completely changed from what we originally wrote in our technical report. Instead of being a trie, our itemset holder eventually became a Dynamic Array of Dynamic Arrays of Itemsets. This was because we wanted fast random access for the Apriori algorithm. Itemset Holder was used both to hold our large itemsets, and temporarily hold our candidates between steps.  
  The main advantage of our itemset holder was that it allowed to us to logically group our data together, and maintain fast access. The biggest disadvantage, however, was that its encapsulation of the data resulted in longer times to pull the data from the holder into the Apriori modules of the program. This increased our time significantly, and in retrospect I wish we could have used pointers more than copying.
* Run Apriori: This is the main function from which our apriori implementation is managed. Ignoring the extra lines related to the timer, it reads very similar to the actual plain text, discrete math version of the algorithm.

# **4.** **Experimentation Detail**

The experiment was conducted on the following system:

Memory on board: 8GB

Processor type: Intel Core i5-4200

CPU speed: 2.80 GHz

System type: Windows 7 SP1 64 bits

Below is the average of all three runs for each of the itemsets that produced any itemsets with minimum support, sorted by minimum support. All itemsets that produced no itemsets are left out from the tables below. If I would have had time (and the ability) to care about this report, I might even have included a nice graph! But I do not, and I have not. Which is too bad.

Dataset: T5.N0.1K.D1K.txt MinSupport: 15

Itemset: Itemsets generated: Avg:

Set 1 5 0.000506334

Dataset: T5.N0.1K.D10K.txt MinSupport: 15

Itemset: Itemsets generated: Avg:

Set 1 4 0.00847304

Dataset: T5.N0.1K.D100K.txt MinSupport: 15

Itemset: Itemsets generated: Avg:

Set 1 5 0.224643

Dataset: T20.N0.1K.D1K.txt MinSupport: 15

Itemset: Itemsets generated: Avg:

Set 1 55 0.000643426

Set 2 196 0.425669

Set 3 68 0.322686

Set 4 3 0.0202284

Dataset: T20.N0.1K.D10K.txt MinSupport: 15

Itemset: Itemsets generated: Avg:

Set 1 55 0.00482288

Set 2 185 3.54709

Set 3 66 2.86251

Set 4 2 0.163873

Dataset: T20.N0.1K.D100K.txt MinSupport: 15

Itemset: Itemsets generated: Avg:

Set 1 55 0.151366

Set 2 184 33.6701

Set 3 66 26.0395

Set 4 2 1.95907

Dataset: T20.N0.5K.D1K.txt MinSupport: 15

Itemset: Itemsets generated: Avg:

Set 1 9 0.00192405

Dataset: T20.N0.5K.D100K.txt MinSupport: 15

Itemset: Itemsets generated: Avg:

Set 1 11 0.729796

Dataset: T25.N0.1K.D1K.txt MinSupport: 15

Itemset: Itemsets generated: Avg:

Set 1 64 0.000534193

Set 2 557 0.604534

Set 3 719 2.76159

Set 4 206 0.916446

Set 5 8 0.0545927

Dataset: T25.N0.1K.D10K.txt MinSupport: 15

Itemset: Itemsets generated: Avg:

Set 1 66 0.00452145

Set 2 555 4.95286

Set 3 700 16.8149

Set 4 206 7.11739

Set 5 12 0.483643

Dataset: T25.N0.1K.D100K.txt MinSupport: 15

Itemset: Itemsets generated: Avg:

Set 1 66 0.152179

Set 2 552 53.8706

Set 3 708 172.935

Set 4 213 73.7329

Set 5 11 5.47456

Dataset: T25.N0.5K.D1K.txt MinSupport: 15

Itemset: Itemsets generated: Avg:

Set 1 25 0.00299878

Dataset: T25.N0.5K.D10K.txt MinSupport: 15

Itemset: Itemsets generated: Avg:

Set 1 25 0.0355183

Dataset: T25.N0.5K.D100K.txt MinSupport: 15

Itemset: Itemsets generated: Avg:

Set 1 26 0.987369

Dataset: T25.N1K.D1K.txt MinSupport: 15

Itemset: Itemsets generated: Avg:

Set 1 1 0.00438925

Dataset: T25.N1K.D10K.txt MinSupport: 15

Itemset: Itemsets generated: Avg:

Set 1 1 0.0920444

Dataset: T25.N1K.D100K.txt MinSupport: 15

Itemset: Itemsets generated: Avg:

Set 1 1 2.28006

Dataset: T20.N0.1K.D1K.txt MinSupport: 25

Itemset: Itemsets generated: Avg:

Set 1 35 0.000514643

Set 2 28 0.153613

Set 3 2 0.0193141

Dataset: T20.N0.1K.D10K.txt MinSupport: 25

Itemset: Itemsets generated: Avg:

Set 1 34 0.00725974

Set 2 33 1.36558

Set 3 3 0.1251

Dataset: T20.N0.1K.D100K.txt MinSupport: 25

Itemset: Itemsets generated: Avg:

Set 1 34 0.173087

Set 2 32 13.7113

Set 3 2 1.17927

Dataset: T25.N0.1K.D1K.txt MinSupport: 25

Itemset: Itemsets generated: Avg:

Set 1 42 0.00072529

Set 2 144 0.224503

Set 3 53 0.19639

Set 4 3 0.0199627

Dataset: T25.N0.1K.D10K.txt MinSupport: 25

Itemset: Itemsets generated: Avg:

Set 1 42 0.00572265

Set 2 148 2.057

Set 3 56 2.08195

Set 4 5 0.145996

Dataset: T25.N0.1K.D100K.txt MinSupport: 25

Itemset: Itemsets generated: Avg:

Set 1 43 0.199678

Set 2 150 24.3853

Set 3 56 23.8752

Set 4 4 1.54862

Dataset: T20.N0.1K.D1K.txt MinSupport: 40

Itemset: Itemsets generated: Avg:

Set 1 10 0.000573903

Set 2 3 0.0229474

Dataset: T20.N0.1K.D10K.txt MinSupport: 40

Itemset: Itemsets generated: Avg:

Set 1 9 0.00642583

Set 2 3 0.0928101

Dataset: T20.N0.1K.D100K.txt MinSupport: 40

Itemset: Itemsets generated: Avg:

Set 1 9 0.183049

Set 2 3 0.946613

Dataset: T25.N0.1K.D1K.txt MinSupport: 40

Itemset: Itemsets generated: Avg:

Set 1 20 0.000743129

Set 2 13 0.0564555

Set 3 1 0.00957502

Dataset: T25.N0.1K.D10K.txt MinSupport: 40

Itemset: Itemsets generated: Avg:

Set 1 19 0.0104218

Set 2 13 0.480435

Set 3 1 0.052563

Dataset: T25.N0.1K.D100K.txt MinSupport: 40

Itemset: Itemsets generated: Avg:

Set 1 19 0.246736

Set 2 13 5.0621

Set 3 1 0.581219

# **5.** **Discussion and Conclusion**

The number of distinct itemsseem to have a surprisingly low effect on the number of large itemsets, as in most cases the difference between the generated number of itemsets only slightly differ when the number of items is the only variable to change.

The average transaction length has a reasonably large effect on the number of generated large itemsets, as none of the average transactions of 5 generated more than, at most, 1-itemsets, while the length of 20 produced at most 4-itemsets, and 25 produced at most 5-itemsets.

The impact of an increase of transactions had a large impact on the number of large itemsets generated, but possibly the opposite way of what might be expected. On average, when the size of the transaction increase the number bigger itemsets are less present. This is due to the minimum support being a percentage of the number of transactions, so changing the number of transactions changes the amount of items present in the transaction list for them to reach minimum support.

The increase of minimum support, without exception, decrease the number of generated large itemsets. This is very unsurprising.

Our data structure seems to be quite efficient. At first we wanted to use a trie structure, but after having done some testing, we decided the current structure was a lot more efficient. This was due to the way we handled getting itemsets at a certain level, where for the trie we had to traverse the trie for every single itemset, while our current structure allows for, at least partially, random access using a dynamic array.

# **6.** **Reference**

[1] R. Agrawal and R. Srikant, “ Fast Algorithms for Mining Association Rules in Large Databases,” *Proceedings of the 20th International Conference on Very Large Data Bases*, pp. 487–499, 1994.

[2] W. K. Chen, “Data structures for selective association mining,” 2005.

# **7.** **Appendix**

Dataset: T5.N0.1K.D1K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 5 0.000515743 0.000493383 0.000509878 0.000506334

Dataset: T5.N0.1K.D1K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.000395146 0.000429236 0.000421171 0.000415184

Dataset: T5.N0.1K.D1K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.000450862 0.000756203 0.000474688 0.000560584

Dataset: T5.N0.1K.D10K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 4 0.0100322 0.00594259 0.00944428 0.00847304

Dataset: T5.N0.1K.D10K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.00665077 0.0047982 0.00707121 0.00617339

Dataset: T5.N0.1K.D10K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.00748725 0.00619734 0.0109761 0.00822024

Dataset: T5.N0.1K.D100K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 5 0.231121 0.207341 0.235467 0.224643

Dataset: T5.N0.1K.D100K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.204152 0.197997 0.221081 0.207743

Dataset: T5.N0.1K.D100K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.205108 0.204736 0.165767 0.191871

Dataset: T5.N0.5K.D1K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.00231736 0.00136652 0.00220519 0.00196302

Dataset: T5.N0.5K.D1K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.00180235 0.00246985 0.00313258 0.00246826

Dataset: T5.N0.5K.D1K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.00247571 0.00214361 0.00231919 0.00231284

Dataset: T5.N0.5K.D10K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.0401491 0.0450173 0.0480865 0.0444176

Dataset: T5.N0.5K.D10K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.0292918 0.0438253 0.0398709 0.0376627

Dataset: T5.N0.5K.D10K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.0351302 0.0250012 0.0254396 0.0285237

Dataset: T5.N0.5K.D100K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.916436 0.749971 1.02652 0.897641

Dataset: T5.N0.5K.D100K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.759989 1.00301 0.81551 0.859503

Dataset: T5.N0.5K.D100K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.718138 0.693826 0.614884 0.675616

Dataset: T5.N1K.D1K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.00530661 0.00378688 0.00339686 0.00416345

Dataset: T5.N1K.D1K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.00367764 0.00271837 0.00269821 0.00303141

Dataset: T5.N1K.D1K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.00419009 0.00564165 0.00983723 0.00655632

Dataset: T5.N1K.D10K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.0706872 0.0581378 0.0590252 0.0626167

Dataset: T5.N1K.D10K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.0657068 0.0547765 0.0580058 0.0594964

Dataset: T5.N1K.D10K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.0650422 0.068037 0.0933142 0.0754645

Dataset: T5.N1K.D100K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 1.98567 1.92926 2.01703 1.97732

Dataset: T5.N1K.D100K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 1.8369 1.98116 1.7704 1.86282

Dataset: T5.N1K.D100K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 1.97592 1.8705 2.03417 1.9602

Dataset: T20.N0.1K.D1K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 55 0.000945711 0.000494482 0.000490084 0.000643426

Set 2 196 0.402645 0.438404 0.435958 0.425669

Set 3 68 0.316298 0.298402 0.353359 0.322686

Set 4 3 0.0210501 0.0184388 0.0211964 0.0202284

Dataset: T20.N0.1K.D1K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 35 0.000522707 0.000501447 0.000519775 0.000514643

Set 2 28 0.164429 0.147314 0.149096 0.153613

Set 3 2 0.0165668 0.0176599 0.0237157 0.0193141

Dataset: T20.N0.1K.D1K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 10 0.000648802 0.000541401 0.000531504 0.000573903

Set 2 3 0.031623 0.0193666 0.0178527 0.0229474

Dataset: T20.N0.1K.D10K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 55 0.00503316 0.004476 0.00495949 0.00482288

Set 2 185 3.46691 3.71216 3.46219 3.54709

Set 3 66 2.63578 3.37436 2.57739 2.86251

Set 4 2 0.157855 0.181761 0.152004 0.163873

Dataset: T20.N0.1K.D10K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 34 0.00915691 0.00497965 0.00764267 0.00725974

Set 2 33 1.29726 1.37306 1.42643 1.36558

Set 3 3 0.104294 0.142898 0.128109 0.1251

Dataset: T20.N0.1K.D10K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 9 0.00751254 0.00574941 0.00601553 0.00642583

Set 2 3 0.0962151 0.0938798 0.0883353 0.0928101

Dataset: T20.N0.1K.D100K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 55 0.152721 0.158625 0.142752 0.151366

Set 2 184 34.4627 33.7126 32.835 33.6701

Set 3 66 26.4701 25.9608 25.6876 26.0395

Set 4 2 1.93009 1.97282 1.97431 1.95907

Dataset: T20.N0.1K.D100K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 34 0.169993 0.168888 0.180379 0.173087

Set 2 32 13.2743 14.0798 13.7799 13.7113

Set 3 2 1.14731 1.27035 1.12015 1.17927

Dataset: T20.N0.1K.D100K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 9 0.168389 0.189138 0.191619 0.183049

Set 2 3 0.934664 0.957213 0.947961 0.946613

Dataset: T20.N0.5K.D1K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 9 0.00154136 0.00156885 0.00266192 0.00192405

Dataset: T20.N0.5K.D1K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.00166599 0.00148455 0.00168835 0.00161296

Dataset: T20.N0.5K.D1K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.00148015 0.00148491 0.00154026 0.00150178

Dataset: T20.N0.5K.D10K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 9 0.0284644 0.0285084 0.0265529 0.0278419

Dataset: T20.N0.5K.D10K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.0348832 0.0284065 0.0382793 0.0338563

Dataset: T20.N0.5K.D10K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.0347197 0.0326142 0.0294245 0.0322528

Dataset: T20.N0.5K.D100K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 11 0.703529 0.744008 0.741851 0.729796

Dataset: T20.N0.5K.D100K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.766759 0.791435 0.697134 0.751776

Dataset: T20.N0.5K.D100K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.750572 0.788885 0.830733 0.790063

Dataset: T20.N1K.D1K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.00310802 0.0040475 0.0085114 0.00522231

Dataset: T20.N1K.D1K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.00468824 0.00312378 0.00310252 0.00363818

Dataset: T20.N1K.D1K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.00302334 0.00379458 0.0053539 0.00405727

Dataset: T20.N1K.D10K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.0566188 0.0508543 0.0546738 0.054049

Dataset: T20.N1K.D10K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.0635962 0.0924389 0.0638286 0.0732879

Dataset: T20.N1K.D10K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.057761 0.0789416 0.0733146 0.0700057

Dataset: T20.N1K.D100K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 1.75604 1.82766 1.78741 1.79037

Dataset: T20.N1K.D100K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 1.72903 1.90185 1.6687 1.76652

Dataset: T20.N1K.D100K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 1.86774 1.73528 1.84732 1.81678

Dataset: T25.N0.1K.D1K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 64 0.000648435 0.000460393 0.000493749 0.000534193

Set 2 557 0.608199 0.625961 0.579441 0.604534

Set 3 719 2.75145 2.7279 2.8054 2.76159

Set 4 206 0.965299 0.920013 0.864026 0.916446

Set 5 8 0.0642505 0.0495472 0.0499805 0.0545927

Dataset: T25.N0.1K.D1K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 42 0.000532971 0.000747772 0.000895127 0.00072529

Set 2 144 0.209061 0.235415 0.229032 0.224503

Set 3 53 0.181898 0.201538 0.205733 0.19639

Set 4 3 0.0169752 0.0251061 0.0178069 0.0199627

Dataset: T25.N0.1K.D1K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 20 0.000777463 0.000764267 0.000687657 0.000743129

Set 2 13 0.0505607 0.0629543 0.0558516 0.0564555

Set 3 1 0.00987792 0.00923645 0.0096107 0.00957502

Dataset: T25.N0.1K.D10K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 66 0.00486785 0.00414757 0.00454895 0.00452145

Set 2 555 4.94201 4.97976 4.93682 4.95286

Set 3 700 16.5095 16.6144 17.3208 16.8149

Set 4 206 7.23279 6.799 7.32039 7.11739

Set 5 12 0.460653 0.518896 0.47138 0.483643

Dataset: T25.N0.1K.D10K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 42 0.00634323 0.0053539 0.00547083 0.00572265

Set 2 148 2.05003 2.05798 2.06299 2.057

Set 3 56 2.087 2.10114 2.05769 2.08195

Set 4 5 0.158362 0.133865 0.145763 0.145996

Dataset: T25.N0.1K.D10K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 19 0.00674461 0.0147304 0.00979031 0.0104218

Set 2 13 0.434468 0.486719 0.520117 0.480435

Set 3 1 0.0540217 0.0506542 0.053013 0.052563

Dataset: T25.N0.1K.D100K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 66 0.121033 0.157996 0.177509 0.152179

Set 2 552 54.5762 53.5332 53.5024 53.8706

Set 3 708 174.493 172.405 171.906 172.935

Set 4 213 74.0447 73.5 73.654 73.7329

Set 5 11 5.39761 5.46247 5.5636 5.47456

Dataset: T25.N0.1K.D100K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 43 0.202876 0.168119 0.22804 0.199678

Set 2 150 24.4983 24.1965 24.4612 24.3853

Set 3 56 23.4106 23.3649 24.8501 23.8752

Set 4 4 1.56719 1.58089 1.49778 1.54862

Dataset: T25.N0.1K.D100K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 19 0.255084 0.252822 0.232301 0.246736

Set 2 13 5.09894 4.95294 5.13443 5.0621

Set 3 1 0.67686 0.51671 0.550087 0.581219

Dataset: T25.N0.5K.D1K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 25 0.00426523 0.00282247 0.00190865 0.00299878

Dataset: T25.N0.5K.D1K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.00238224 0.00414427 0.00389464 0.00347372

Dataset: T25.N0.5K.D1K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.0016484 0.00309262 0.0020725 0.00227117

Dataset: T25.N0.5K.D10K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 25 0.0316003 0.0318888 0.0430658 0.0355183

Dataset: T25.N0.5K.D10K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.0315211 0.0256031 0.0333796 0.0301679

Dataset: T25.N0.5K.D10K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.034131 0.0366104 0.0309816 0.0339077

Dataset: T25.N0.5K.D100K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 26 0.96149 1.00227 0.998346 0.987369

Dataset: T25.N0.5K.D100K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.991394 1.01441 0.921777 0.975861

Dataset: T25.N0.5K.D100K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.910675 1.01142 1.02001 0.980703

Dataset: T25.N1K.D1K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 1 0.00339686 0.00454491 0.00522597 0.00438925

Dataset: T25.N1K.D1K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.00665077 0.00461016 0.00745572 0.00623888

Dataset: T25.N1K.D1K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.0105172 0.00678383 0.00710676 0.00813593

Dataset: T25.N1K.D10K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 1 0.0806688 0.11188 0.083584 0.0920444

Dataset: T25.N1K.D10K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.097045 0.0860597 0.0908502 0.0913183

Dataset: T25.N1K.D10K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 0.0912535 0.102343 0.0814562 0.0916842

Dataset: T25.N1K.D100K.txt MinSupport: 15

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 1 2.30908 2.23389 2.29723 2.28006

Dataset: T25.N1K.D100K.txt MinSupport: 25

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 2.18688 2.35972 2.19587 2.24749

Dataset: T25.N1K.D100K.txt MinSupport: 40

Itemset: Itemsets generated: Run 1: Run 2: Run 3: Avg:

Set 1 0 2.33459 2.29636 2.17234 2.26776

[1] To check the CPU speed, in Windows 7 and Vista, click **Start**->**Computer**, right-click on **Properties**, and all the needed information are available in the pop-up window

[2] A timer system is given to you on Canvas