# Frequent Item set Mining Using Data Grid in WEKA 3.8

**Akanksha[1] Dr. Kanwal Garg[2]**
[1,2]Department of Computer Science & Applications Engineering
[1,2]Kurukshetra University, Kurukshetra-136119

*Abstract*— The premise of this paper is to generate the frequent itemsets by making use of data grids in WEKA 3.8 environment. The dynamic nature grid environment allows the most secure form of extraction of frequent itemsets while data grids are used to create data and validate them. WEKA 3.8 is easy to use and implement tool and it provides various algorithms to accomplish best rules to make obvious decisions.

***Key words:*** WEKA 3.8, Apriori Algorithm, Data Grids, Frequent Itemset Mining, Visualization Tools

## I. INTRODUCTION

The exponential growth of technology used in social, business, and scientific domain, database sizes has made difficult to interpret meanings out of generated data. In this process, data mining plays vital role in automatically extracting useful and hidden information from such large databases. The main idea of data mining is to find effective ways to combine the computer's power to process data with human eye's ability to detect patterns. The real meaning of data mining is the non-trivial process of recognizing suitable, potentially helpful and finally comprehensible patterns in data. Data mining is the use of statistical methods with computers to reveal helpful patterns in the databases. Huge data is a collection of large and complex datasets which is complicated to process by using conventional methods and offered technologies of data mining[1].

Now a day the processor is having speed that is underutilized due to the improper localization of the various parameters if proper localization of parameters is done then the performance of processor can be improved. This can be done using several cache conscious mechanisms that are going to help in optimal use of the resources for better outcome. Here the researcher is using WEKA tool for the mining of frequent patterns[2].

## II. DATA GRID IN WEKA 3.8

Grid computing is solving such large scale of problems, which could not be solved within the traditional computing methods due to the limited memory, or computing power. Grid system required a high-speed network working under specific or specialized software called grid middleware, which allows the distributed resources to work together in a relatively smooth and transparent manner. Grids emerged as effective infrastructures for distributed high-performance computing and data processing, a few Grid-based KDD systems has been proposed. Grid technology delivers high performance and manage data and knowledge distribution. Because scalable knowledge discovery is critical on distributed data mining services beginning to allow distributed teams or virtual organizations accessing and data mining in high level, standard and reliable way[3].

This paper presents Weka 3.8, a framework that extends the widely used Weka toolkit for supporting distributed data mining on Grid environments. Weka provides a large collection of machine learning algorithms written in Java for data pre-processing, classification, clustering, association rules and visualization, which can be invoked through a common Graphical User Interface (GUI)[4].

In Weka3.8, the data-preprocessing and visualization phases are still executed locally, whereas data mining algorithms for classification, clustering and association rules can be also executed on remote Grid resources. To enable remote invocation, each data mining algorithm provided by the Weka library is exposed as a Web Service, which can be easily deployed on the available Grid nodes. Thus, Weka 3.8 also extends the Weka GUI to enable the invocation of the data mining algorithms that are exposed as Web Services on remote machines. To achieve integration and interoperability with standard Grid environments, Weka3.8 has been designed and developed by using the emerging Web Services Resource Framework (WSRF) as enabling technology[5].

## III. WEKA 3.8

WEKA 3.8 is a data mining system developed by the University of Waikato in New Zealand that implements data mining algorithms. WEKA 3.8 is a state-of-the-art facility for developing machine learning (ML) techniques and their application to real-world data mining problems. It is a collection of machine learning algorithms for data mining tasks. The algorithms are applied directly to a dataset. WEKA 3.8 implements algorithms for data preprocessing, classification, regression, clustering, association rules; it also includes a visualization tools. The key features of WEKA 3.8 are it provides many different algorithms for data mining and machine learning and it is open source and freely available and platform-independent and it provides flexible facilities for scripting experiments. The processing steps in WEKA 3.8 includes first we collect the sample data min Excel file formats and after that it is converted into .csv format and after it is again converted into. arff format that one is used in WEKA 3.8 explorer to analysis Apriori algorithm [6].

## IV. APRIORI ALGORITHM

The author proposed an algorithm for frequent itemset mining and association rule learning over transactional databases called Apriori. The Apriori algorithm is an influential algorithm for mining frequent itemsets for boolean association rules[7]. The main key factors of Apriori algorithm are below: -

1) Frequent Itemsets: The sets of item which has minimum support (denoted by Li for ith Itemset).
2) Apriori Property: Any subset of frequent itemset must be frequent.
3) Join Operation: To find Lk, a set of candidate k-itemsets is generated by joining Lk-1 with itself [8].

The frequency of an item set is computed by counting its occurrence in each transaction. Apriori is a significant algorithm for mining frequent itemsets for Boolean association rules. Apriori is an iterative level wise search algorithm, where k- itemsets are used to explore (k+1)-itemsets. First, the set of frequents 1-itemsets is found. This set is denoted by L1. L1 is used to find L2, the set of frequent 2-itemsets, which is used to find L3 and so on, until no more frequent k-itemsets can be found. The finding of each Lk requires one full scan of database.

The two main steps of Apriori algorithm are: -

- The Join step: - To find Lk a step of candidate k-itemsets is generated by joining Lk-1 with itself. This set of candidate is denoted by Ck.
- The Prune step: - Ck is a superset of Lk, that is, its members may or may not be frequent, but all of the frequent k-itemsets are included in Ck.
- To reduce the size of Ck, the Apriori property is used as follows: -
- Any (k-1) itemset that is not frequent cannot be a subset of frequent k-itemset.
- Hence, if (k-1) subset of candidate k itemset is not in Lk-1 then the candidate cannot be frequent either and so can be removed from C [7].

Apriori algorithm pseudo code: -

```
 procedure Apriori(T, minSupport)
{  // T is the database and minSupportis the minimum support
L1= {frequent items};
for(k= 2; Lk-1 !=∅; k++)
{
Ck= candidates generated from Lk-1
//that is cartesian product Lk-1 x Lk-1 and
//eliminating any k-1 size itemset that is not
//frequent
foreachtransaction t in database do
{
#increment the count of all candidates in Ck
that are contained in t
Lk = candidates in Ck with minSupport
}//end for each
}//end for
return;
}
```

## V. Research Methodology

The main methodology used in present research work is data grids in WEKA 3.8 by implementing the Apriori Algorithm for the study and analysis of frequent itemset mining.

In this paper the data is obtained from National Institute of Statics (NIS) for the region of Flanders. More specifically the data are obtained from the Belgian "Analysis Form for Traffic Accidents" that should be filled out by a police officer for each traffic accident that occurs with injured or deadly wounded casualties on a public road in Belgium. In total, 340,184 traffic accident records are included in the dataset.

The traffic accident data contain a rich source of information on the different circumstances in which the accidents have occurred: course of the accident (type of collision, road users, injuries …), traffic conditions (maximum speed, priority regulation …), environmental conditions (weather, light conditions, time of the accident …), road conditions (road surface, obstacles …), human conditions (fatigue, alcohol …) and geographical conditions (locations, physical characteristics …). The attributes are retrieved fromhttp://fimi.ua.ac.be/data/accidents.pdf.

## VI. Experimental Results

For test the researcher considers 45 different attributes that have taken in the traffic accidents records. In the experimental results the researcher has demonstrated use of Apriori algorithm for frequent itemset mining using WEKA 3.8.The emphasis in this study lies on the identification and profiling of frequently occurring accident patterns at high frequency accident locations and the degree in which these accident characteristics are discriminating between high frequency and low frequency accident locations. Selecting the association rules that appear in both the high frequency accident rules set and the low frequency accident rules set results.

Using the Apriori algorithm the researcher wants to find the association rules that have minSupport=50%and minConfidence=50%. We will do this using WEKA GUI.

After the researcher launch the WEKA application and open the accident.csv.arffas shown in Figure 1 then move to Associate tab and set up the configuration as shown in Figure 2.

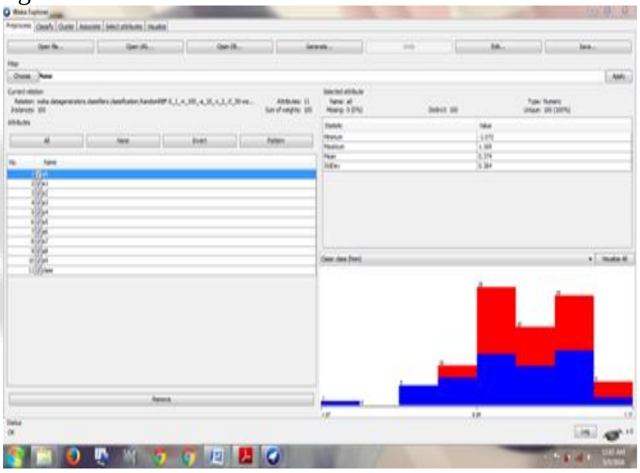
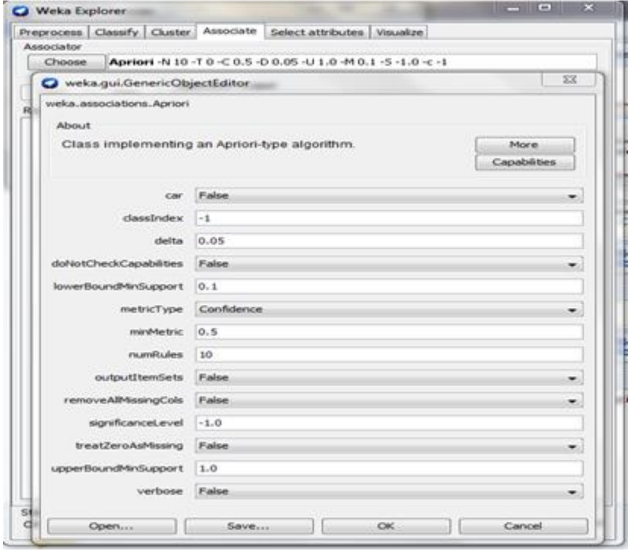


Fig. 1: WEKA 3.8 GUI



Apriori -N 10 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

weka.associations.Apriori

About

Class implementing an Apriori-type algorithm.

| | |
|---|---|
| car | False |
| classIndex | -1 |
| delta | 0.05 |
| doNotCheckCapabilities | False |
| lowerBoundMinSupport | 0.1 |
| metricType | Confidence |
| minMetric | 0.5 |
| numRules | 10 |
| outputItemSets | False |
| removeAllMissingCols | False |
| significanceLevel | -1.0 |
| treatZeroAsMissing | False |
| upperBoundMinSupport | 1.0 |
| verbose | False |

Fig. 2: Associate Tab

## VII. Analysis & interpretation

After the configuration set up in Associate tab, the best results of Apriori algorithm are shown in the Figure 3 in which the minConfidence is set to 0.5 and minSupport is set to 0.35 the whole process is performed in 13 cycles on the full dataset. Ten best rules found for each class with different values of Confidence, Lift, Leverage, and Conviction. After the algorithm is finished, the results are following: -

Run Information
Scheme: weka.associations.Apriori-N 10 to C-.5 D-.05 U 1.0 M-.1-S-1.0-c-1
Relation: accidents.csv.arff
Instances: 44127
Attributes: 45
Associator Model: Full training set
Minimum Support: .35
Minimum Metric<confidence>: .5
Number of cycles performed: 13
Best rules found:

1) a5 = false 44==> class = c0 42 <conf: (0.95)> lift: (1.45) lev: (0.13) [12] conv: (4.99)
2) a8 = true 48==> class = c0 36 <conf: (0.75)> lift: (1.14) lev: (0.04) [4] conv: (1.26)
3) a2 = false 58==> class = c0 41 <conf: (0.71)> lift: (1.07) lev: (0.03) [2] conv: (1.1)
4) a7 = false50==> class = c0 35 <conf: (0.7)> lift: (1.06) lev: (0.02) [1] conv: (1.06)
5) a4 = false 54==> a0 = true 36 <conf: (0.67)> lift: (1.23) lev: (0.07) [6] conv: (1.31)
6) a0 = true 54==> a4 = false 36 <conf: (0.67)> lift: (1.23) lev: (0.07) [6] conv: (1.31)
7) a0 = true 54==> class = c0 36 <conf: (0.67)> lift: (1.01) lev: (0.0) [0] conv: (0.97)
8) a5 = true 56==> a3 = true 37 <conf: (0.66)> lift: (1.08) lev: (0.03) [2] conv: (1.09)
9) a4 = false 54==> class = c0 35 <conf: (0.65)> lift: (1.98) lev: (-0.01) [0] conv: (0.92)
10) class = c 66==> a5 = false 42 <conf: (0.64)> lift: (1.45) lev: (0.13) [12] conv: (1.48)

The best rules are shown in Figure 3 given below:

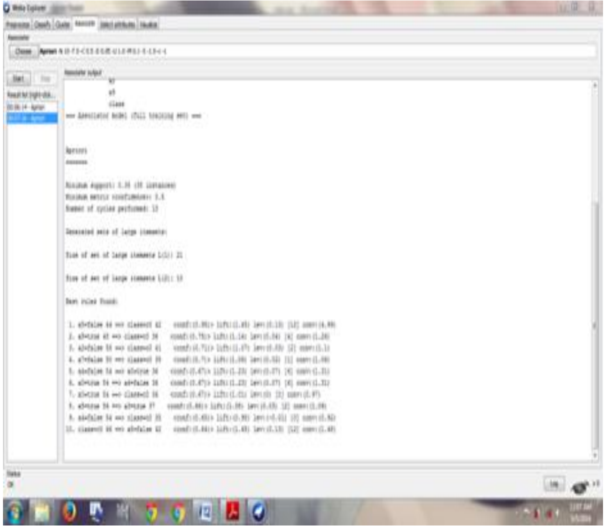


Fig. 3: Best Rules

## VIII. Conclusion

In this paper, the Apriori algorithm with WEKA tool was used on a large dataset of traffic accidents in terms of accident related data and location characteristics. The analysis showed that by generating association rules the identification of accident circumstances that frequently occur together is facilitated. This leads to the strong understanding of the occurrence of traffic accidents. In this paper revealed several interesting rules which in turn provide valuable input for purposive government traffic safety actions which will help in reduce in traffic accidents in future.

## References

[1] J. Han, M. Kamber and J. Pei, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2006.

[2] R. Agarwal, T. I. Ski and A. Swami, "Mining Association Rules Between Sets of Items in Large Databases," in ACM SIGMOD International Conference on Management of Data, New York, 1993.

[3] V. Crucin, M. Ghanem, Y. Guo, M. Kohler, A. Rowe and P. Wendel, "Discovery Net: Towards a Grid of Knowledge Discovery," in Knowledge Discovery and Data Mining, 2002.

[4] H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools with Java Implementations, Morgan Kaufmann, 2000.

[5] K. e. a. Czajkowski, "The WS Resource Framework Version 1.0," 01 05 2004. [Online]. Available: toolkit.globus.org/wsrf/specs/ws-wsrf.pdf. [Accessed 22 05 2016].

[6] K. R. Swamy and G. H. Babu, "Identification of Frequent Item Search Patterns Using Apriori Algorithm and WEKA Tool," International Journal of Innovative Technology and Research, vol. 3, no. 5, pp. 2401-2403, 2015.

[7] R. Agarwal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," in International Conference on Very Large Databases, 1994.

[8] P. Tanna and D. Y. Ghodasara, "Using Apriori with WEKA for Frequent Pattern Mining," Internatioanl Journal of Trends and Technology, vol. 12, 2014.