Integration of Clustering and Association Rule Mining with Prioritizing

M.T. Dulaj (mtdulajbangalawaththa@gmail.com) And E.Y. A. Charles.
Department of Computer Science, University of Jaffna.

1. INTRODUCTION

- Clustering and Association are two important techniques of data mining.
- Association rule learning is a well-researched method for discovering interesting relations between variables in large datasets. It identifies and defines strong rules discovered in datasets using different measures of interestingness.
- Clustering is an unsupervised learning problem that group objects based upon distance or similarity. Each group formed is known as a cluster.
- In this poster we make use of a large dataset 'Nursery Dataset' from UCI (University of California Irvine) machine learning repository, containing 8 attributes and 12960 instances to perform an integration of clustering and association rule mining with prioritizing to determine some essential and interesting rules formed in each clusters.
- Clustering and association rule mining were performed using WEKA (Waikato Environment for Knowledge Analysis), a Data Mining tool.
- Proposed prioritization and validation processes was applied to rearrange the obtained rules based on their importance.
- The final results of the experiment show that integration of clustering and Association Rule Mining with prioritizing give some essential and well defined rules with an important order.

2. DATA SET

In this work I have used an existing dataset, Nursery Dataset from UCI machine learning repository. The nursery dataset is a multivariate dataset with categorical values. It contains 8 attributes and 12960 instances. It does not contain any missing values and it consists attribute values as shown in following table.

Attribute	Description	Values
Parents	Parent's occupation	usual, pretentious, great_pret
Has_nurs	Child's nursery	proper, less_proper, improper, critical, very_crit
Form	Form of the family	complete, completed, incomplete, foster
Children	Number of children	1, 2, 3, more
Housing	Housing condition	convenient, less_conv, critical
Finance	Financial standing of the family	convenient, inconv
Social	Social condition	non_prob, slightly_prob, problematic
Health	Health condition	recommended, priority, not_recom

The class attribute (Evaluation) describe evaluation of applications for nursery school and it contains 5 values as follows.

Class	N (Number of instances)	N[%]
not_recom	4320	33.333 %
Recommend	2	0.015 %
very_recom	328	2.531 %
priority	4266	32.917 %
spec_prior	4044	31.204 %

3. OBJECTIVES

- To identify hidden association rules in large data sets.
- To structure the rules and clusters by order of priority.
- To determine and validate important association rules in data set.

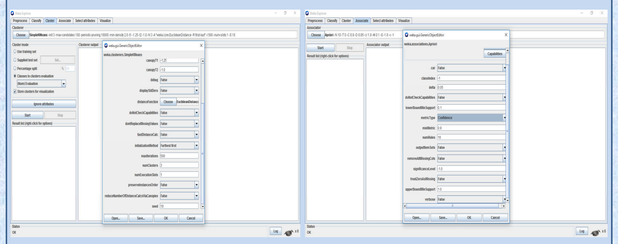
4. METHODOLOGY

Begin • Data Set
1 • Data Preparation
2 • Clustering
3 • Formation of Clusters
4 • Segment Data Set
5 • Association Rule Mining
6 • Rules for Clusters
7 • Prioritizing
8 • Validation
9 • Final Rule Set

5. EXPERIMANTAL SETUP

- Test using WEKA tool.
- Simple KMeans for Clustering.
- Apriori algorithm for ARM.
- Algorithm for prioritizing.
- Conditions for validation.

5.1 CLUSTERING & ASSOCIATION



Basic set up for SimpleKMeans and Apriori

5.2 PRIORITIZING

1. Begin
2. K = Number of clusters.
3. Weight = {Define weight for each class attribute}
4. Clusters = {Evaluated clusters according to class attribute}
5. i = 0; j = 0;
6. while(i < k)
   {
   $c[\,i\,] = \frac{CorrectlyClustered\_in\_Clusters(i)}{Size\ of\ clusters(i)} * \text{clusters(i).class.weight}$
   i += 1
   }
7. Clusters = {Rearrange order of clusters by ascending order of c}
8. While (j < k)
   {
   RulesC (j) = { Association rules over Clusters (j) }
   If(RHS(RulesC(j)) == class)
   {
   Set those rules with high priority than other.
   }
   J += 1
   Return RulesC (j)
   }
9. end

5.3 VALIDATION

- Remove rules that R.H.S contains the class attribute.
- Remove rules with same attributes:
  If Rule1 (LHS U RHS) == Rule2 (LHS U RHS), then remove rule with less priority.

6. EXPERIMENTAL RESULTS

```
Time taken to build model (full training data) : 0.08 seconds
=== Model and evaluation on training set ===

Clustered Instances
0      5677 ( 44%)
1      4245 ( 33%)
2      3038 ( 23%)

Class attribute: Evaluation
Classes to Clusters:

   0    1    2  <-- assigned to cluster
1434 1085 1801 | not_recom
   0    2    0 | recommend
 106  154   68 | very_recom
1945 1645  676 | priority
2192 1359  493 | spec_prior

Cluster0 <-- spec_prior
Cluster1 <-- priority
Cluster2 <-- not_recom

Incorrectly clustered instances :      7322.0      56.4969 %
```

Fig 01 **- Results For K-Means Clustering**
Test mode: Classes to clusters evaluation on training data

Best rules found:

1. Evaluation=not_recom 1801 ==> health=not_recom 1801 <conf:(1)> lift:(1.69) lev:(0.24) [733] conv:(733.32)
2. health=not_recom 1801 ==> Evaluation=not_recom 1801 <conf:(1)> lift:(1.69) lev:(0.24) [733] conv:(733.32)
3. finance=convenient Evaluation=not_recom 1144 ==> health=not_recom 1144 <conf:(1)> lift:(1.69) lev:(0.15) [465] conv:(465.81)
4. finance=convenient health=not_recom 1144 ==> Evaluation=not_recom 1144 <conf:(1)> lift:(1.69) lev:(0.15) [465] conv:(465.81)
5. housing=less_conv Evaluation=not_recom 961 ==> health=not_recom 961 <conf:(1)> lift:(1.69) lev:(0.13) [391] conv:(391.3)
6. housing=less_conv health=not_recom 961 ==> Evaluation=not_recom 961 <conf:(1)> lift:(1.69) lev:(0.13) [391] conv:(391.3)
7. social=slightly_prob Evaluation=not_recom 961 ==> health=not_recom 961 <conf:(1)> lift:(1.69) lev:(0.13) [391] conv:(391.3)
8. social=slightly_prob health=not_recom 961 ==> Evaluation=not_recom 961 <conf:(1)> lift:(1.69) lev:(0.13) [391] conv:(391.3)
9. parents=usual Evaluation=not_recom 753 ==> health=not_recom 753 <conf:(1)> lift:(1.69) lev:(0.1) [306] conv:(306.6)
10. parents=usual health=not_recom 753 ==> Evaluation=not_recom 753 <conf:(1)> lift:(1.69) lev:(0.1) [306] conv:(306.6)

Fig 02 **- Association Rule Mining Results With Apriori For Cluster 2**

EXPERIMENTAL RESULTS Cont.

Prioritizing:

❑ Defining frequency based weight:

$$\text{weight} = \frac{no.of\ element\ in\ class}{total\ instances}$$

not_recom = 4320/12960 = 0.333
Recommend = 2/12960 = 0.0001
very_recom = 328/12960 = 0.025
Priority = 4266/12960 = 0.329
spec_prior = 4044/12960 = 0.312

❑ Finding value c for prioritize clusters:

$$c[\,i\,] = \frac{CorrectlyClustered\_in\_Clusters(i)}{Size\ of\ clusters(i)} * \text{clusters(i).class.weight}$$

c(0) = (2192/5677)*0.312 = 0.1204
c(1) = (1645/4245)*0.329 = 0.1274
c(2) = (1801/3038)*0.333 = 0.1974

❑ Order of prioritize clusters:

Cluster2 > Cluster1 > Cluster0

2. health=not_recom 1801 ==> Evaluation=not_recom 1801
4. finance=convenient health=not_recom 1144 ==> Evaluation=not_recom 1144
6. housing=less_conv health=not_recom 961 ==> Evaluation=not_recom 961
8. social=slightly_prob health=not_recom 961 ==> Evaluation=not_recom 961
10. parents=usual health=not_recom 753 ==> Evaluation=not_recom 753

1. Evaluation=not_recom 1801 ==> health=not_recom 1801
3. finance=convenient Evaluation=not_recom 1144 ==> health=not_recom 1144
5. housing=less_conv Evaluation=not_recom 961 ==> health=not_recom 961
7. social=slightly_prob Evaluation=not_recom 961 ==> health=not_recom 961
9. parents=usual Evaluation=not_recom 753 ==> health=not_recom 753

Fig 03 **- Prioritized rules for cluster2**
(Expected Results from Algorithm)

Validation:

```
// For Cluster 2
 2. health=not_recom 1801 ==> Evaluation=not_recom 1801
 4. finance=convenient health=not_recom 1144 ==> Evaluation=not_recom 1144
 6. housing=less_conv health=not_recom 961 ==> Evaluation=not_recom 961
 8. social=slightly_prob health=not_recom 961 ==> Evaluation=not_recom 961
10. parents=usual health=not_recom 753 ==> Evaluation=not_recom 753

// For Cluster 1
 2. health=not_recom 1085 ==> Evaluation=not_recom 1085
 4. finance=convenient health=not_recom 720 ==> Evaluation=not_recom 720
 6. housing=convenient health=not_recom 645 ==> Evaluation=not_recom 645
 8. social=nonprob health=not_recom 645 ==> Evaluation=not_recom 645
10. parents=pretentious health=not_recom 621 ==> Evaluation=not_recom 621

// For Cluster 0
 2. health=not_recom 1434 ==> Evaluation=not_recom 1434
 4. finance=inconv health=not_recom 1138 ==> Evaluation=not_recom 1138
 6. housing=critical health=not_recom 784 ==> Evaluation=not_recom 784
 8. social=problematic health=not_recom 784 ==> Evaluation=not_recom 784
10. form=incomplete health=not_recom 582 ==> Evaluation=not_recom 582
```

Fig 04 **– Validated Rules for clusters**
(Expected Results)

7. CONCLUSION & FUTURE WORK

The presented experiment shows that integration of clustering and association rule mining give well-defined rules in case of each cluster formed for nursery dataset on WEKA. By clustering, it builds the clusters according to the class attribute in dataset, and after clusters association is applied to demonstrate the rules formed for each clusters. Prioritization process is applied to arrange rules with an important order, and finally by the validation it reduce the resulted rule space which helps decision makers to identify information effectively. Furthermore, this method find interesting content-related rules, rather than applying Association rules mining for the whole dataset at once.

Suggested future work are, to implement this methodology as a single system to compare final expected results of this methodology with the final results of implemented system and analyze the complexity of the system. Also, to extend the prioritization algorithm with additional conditions in order to achieve more reliable prioritization order.

8. REFERENCES

- Ritu Ganda et al., "Knowledge Discovery from Database using an Integration of Clustering and Association Rule Mining", In: International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Volume 3, Issue 9, September 2013.
- Duke Hyun Choi, Byeong Seok Ahn, Soung Hie Kim, "Prioritization of association rules in data mining: Multiple criteria decision approach", In: Expert Systems with Applications 29 (2005) 867-878.