Validation of association Rules developed by LRM Algorithm using Clustering Approach

Saravanan M. S.[#], Rama Sree R. J.[*]

[#] *R & D Centre, Bharathiar University, Coimbatore, Tamil Nadu, INDIA.*
Dept. of I.T in Vel Tech Dr. RR & Dr. SR Technical University, Chennai, INDIA.
[*] *Department of Computer Science, Rashtriya Sanskrit Vidyapeetha,*
Tirupati, Andhra Pradesh, INDIA.

[1]saranenadu@yahoo.co.in, [2]rjramasree@yahoo.com

***Abstract* – The association rule mining technique is very popular in the recent research area and as we know this technique is one of the powerful domains of data mining. The association rule mining concept was used in various applications such as healthcare, banking, sales, etc. The association rule mining concept, first time we are going to use in the domain of dyeing process. Hence, the nature of dyeing process is dynamic. So it has more number of event logs for one process, because the process is controlled by different parameters. So, the dependency variable decides the quality of the shade. The little shade different occurs due to lot of dyeing treatment problems. To overcome these difficulties, we have implemented the association rule mining technique in the domain of dyeing process. This paper validates the association rule mining approach that is particularly a new association rule mining algorithm LRM is used for two dyeing processes from Jayabala dyeing unit. This validation is conducted using clustering approach. The clustering is used because of the complexity of the dyeing processes. It reduces the complexity when the event logs are more in numbers.**

***Keywords*—Dyeing process, parameters, clustering, association rule mining, shade.**

I. INTRODUCTION

The association rule mining found to be a promising technique to obtain information about any process underlying an event log or to make implicit knowledge to explicit. To further explore this research area, the LinkRuleMiner (LRM) was proposed. In our previous research paper, we have implement the association rule mining in Emerald Dyeing unit, which has a lot of low frequent events of dyeing process. Many events have frequency of one whereas the highest frequency of an activity in the log is in 1000s. It was seen that the LRM algorithm easily captures high frequent behaviour whereas the low frequent behaviour remains dependent on the support threshold values. The LRM however cannot distinguish noise with the low frequent behavior [1]. The presence of highly low frequent behavior represents flexibility and lack of standardization in the Jayabala dyeing unit or it may represent rare and specialized dyeing process cases pertaining to a unique type of shades.

The low frequent behaviour in any dyeing log adds up to the heterogeneity of the data as it represents more flexibility, uniqueness and less structure. It is also a kind of behavioural pattern that exists in the log and often goes unnoticed and undiscovered. However, whether some behaviour is low frequent or high frequent, if there would be a mechanism to group shades with similar profile, it would present an opportunity to obtain homogeneity in the heterogeneous and less structured dyeing processes [2]. For this purpose, the LRM provides the functionality of clustering. As apparent, this paper focuses to: Develop a mechanism to use Association Rules for clustering different shade (or complications, treatments etc.) groups into one homogeneous group.

The paper begins with the introduction about the clustering in the LRM algorithm and the section 3, validated the LRM algorithm using clustering approach for the Jayabala dyeing unit and the section 4 concludes this chapter by summarizing the importance of clustering in combination with the association rules.

II. APPLICATION OF CLUSTERING IN THE LRM ALGORITHM

In most simple terms, clustering can be understood as making clusters of similar things. It is the process of organizing objects into groups whose members are similar in some way. A cluster is therefore a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters [3]. In the process of clustering the aim is to find homogeneous chunks in data. Clustering is a key area in data mining and knowledge discovery [4], which are activities oriented towards finding non-trivial or hidden patterns in data collected in databases. Commonly, the applications of clustering includes finding common surfing patterns in the set of web users, partitioning different documents based on their content, finding protein sequences in a database, finding groups of customers with similar purchase patterns etc. All these applications aim at finding homogeneous members in a database, and these groups can be further used as a target for understanding behaviour of elements like customers in a supermarket, relationship between documents, protein sequences etc. So that insights can be gained into the patterns that exist in a pool of data.

Clustering process instances in an event log also aim at finding homogeneity in the log. In dyeing domain, processes are dynamic; less structured and involve various disciplines [5]. Every shade represents unique case in terms of complications, body type, responses to treatment procedures etc. In this situation it becomes difficult to find a group of shades that are similar in one way or another. Clustering in the context of LRM refers to finding homogeneous groups of process instances that is shades, which are similar in some way [6]. The LRM provides two choices for clustering: homogeneous group of process instances or cases can be found that satisfy certain association rule, and secondly homogeneous group of process instances or cases can be found on the basis of a particular frequent pattern in the log (itemset) [7] [8]. Clustering in the LRM is in the form of partitioning, where the entire event log is partitioned in two clusters: one which satisfies a particular association rule or frequent itemset and the other which does not satisfy it.

The next section illustrates that the validation of association rule mining algorithm that is LinkRuleMiner can be used to derive treatments of homogeneous shades from a Jayabala dyeing unit's dyeing process using clustering approach.

III. Validation of LRM Algorithm Using Clustering Approach

To illustrate how clustering approach to identify the similar treatments of dyeing process for the same type of shades using association rules, hence it is explained with an example as shown below. For this illustration the Jayabala dyeing unit's hundred shades dyeing process log is used. It consisting of 600 PIs, 262 different ATEs and 10,200 total numbers of ATEs. Each PI refers to a shade and the complication path followed by the treatments. The LRM applied with default parameter values generates 10 association rules.

A. Illustration 1

This first illustration shows that the cluster derived from the DWS process mining algorithm is used as input to association mining algorithms to obtain frequent item sets. In figure 1, it was shown that the cluster R0 is selected. When this cluster is used as an input for the Apriori algorithm, the process model obtained is much simpler than the process model obtained for the complete log. This represents the usefulness of the clustering functionality. It helps in obtaining simpler (as compared to the spaghetti-models) models which provide better insights into the underlying process.

The output of the DWS process mining algorithm has the cluster R0 as shown in the figure 2 and it is used as a input to HM process mining algorithm and it produce a simple process model as compare to the whole log of the dyeing process as shown in the figure 3. At the same time the process mining algorithm HM do not have any discriminant rule and it is depicted in figure 3. These process models were developed for the Jayabala dyeing unit's dyeing processes whole log and cluster R0 (i.e dereived from DWS algorithm) respectively. Hence, it is concluded that the process model for cluster R0 is simpler than the process model for the Jayabala dyeing whole log.

Though the problems like dangling activities and missing connections exist in these process models, these may be eliminated by accordingly varying the parameter settings of the HM algorithm so as to generate only the detailed behaviour and leave the low frequent one. This process model gives us information about the control flow of the shades frequent items from the treatment *PH_Res_abnorm* that eventually also suffer from the treatments *Pre_Treat_Absent* and *C_Iden_same-1st-two-yrs*.

The association rule mining algorithm LRM is used to derive a simple process model for any type of log. Hencce, the Jayabala dyeing unit't dyeing process derived process model as shown in figure 2, which has has whole log of the dyeing process is used as a input to develop a association rule mining process model using LRM association rule mining algorithm. The output of this algorithm is shown in figure 4. The output of the cluster log as shown in figure 3 is also converted into association rule mining process model and it is shown in figure 5. Hence the respective association rule mining models are shown in the figure 4 and 5 respectively for the Jayabala dyeing whole log association rules and cluster R0 association rules.

B. Illustration 2

In this illustration the observed 10 rules of the hundred shades dyeing process *Pre_Treat_Absent* and *PH_Res_abnorm* always occur together in this log and are the most frequently occurring treatments because a rule involving both these treatments acts as the basis for clustering. This homogeneous cluster of process instances satisfying Rule 10 can be obtained by executing the following command on Weka Command Line Interface.

Java weka.associations.Apriori –t d:\shades100.arff

The output of the association rule process model using Apriori algorithm for cluster R0 can be seen in the figure 6. This figure lists all the process instances in the event log and the process instances satisfying this association rule.

The association rule mining algorithms such as Apriori, FPGrowth, H-Mine and LinkRuleMiner is used to generate association rules for simple understanding of dyeing experts. Therefore, these association mining algorithms produce the same process models, except the Apriori algorithm and even the algorithms were different because of the low frequent items property. But the processing time of each algorithm various depends upon the efficiency of the algorithms. These association rule mining process models are shown in the figures 6, 7, 8 and 9 for the Apriori, FPGrowth, H-Mine and LinkRuleMiner respectively.

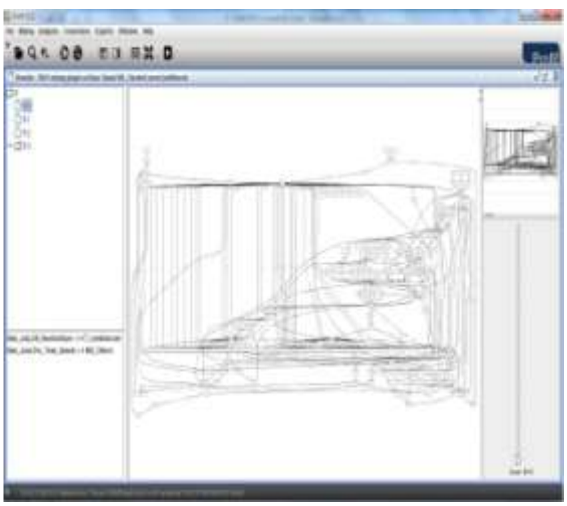

Fig.1 DWS Algorithm Cluster R0

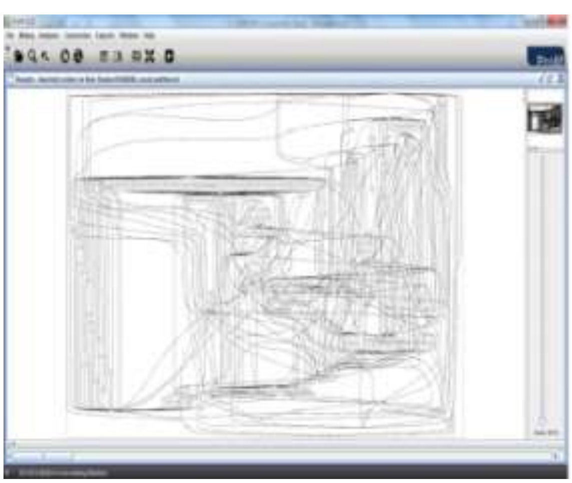

Fig.2 Process Models for the whole log using HM process mining algorithm

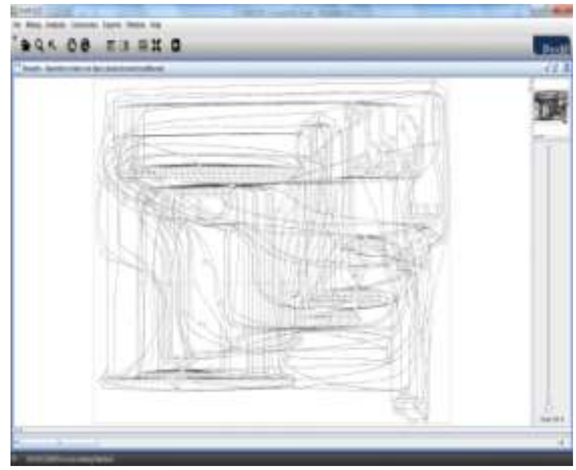

Fig.3 Process Models for the cluster R0 log using HM process mining algorithm

IV. CONCLUSION

The paper discussed that how clustering can be achieved using the LRM algorithm. Two dyeing processes were used in the paper. They showed that how clustering can be performed on the basis of association rules and frequent itemsets respectively. The first illustration showed that the obtained clusters can be used for further mining and the mined process models can be simple and easy to understand. Clusters can also represent frequent patterns existing in the log when the support count of items is chosen as the basis of clustering. The second experiment was conducted using the same log from dyeing process log of Jayabala dyeing, and it was observed that the process models from clustered PIs can be richer in dependencies thereby providing more insights into the underlying process. These clusters may be representative of the highly frequent behavior found in the log or the exceptional dyeing process cases in form of highly low frequent behaviour.

Clustering can therefore be utilized to obtain specific process models or to generate simpler models as opposed to the spaghetti-like models. Further, the clustering functionality in LRM could also be enhanced to obtain a hierarchy of logs based on the association rules. The cluster from an association rule can be further mined with the LRM again, and one of the generated rules can be used for further clustering. Repeating this sequence of generating association rules and clustering can give the user a tree of logs satisfying different variants of the same association rule (that is initially used for clustering). The leaf nodes then would be the logs representing the most basic associations (may be in form of a rule: a=>b, instead of a=>b, c).

```
LinkRuleMiner found 20 rules (displaying top 10)

 1. [Post_Treat_Absent=yes, PH_Neu_normal=yes]: 301 ==> [ShadeCheck_Good=yes]: 297    <conf:(0.99)> lift:(1.17) lev:(0.07) conv:(9.23)
 2. [Pre_Treat_Absent=yes]: 305 ==> [PH_Res_abnorm=yes]: 299    <conf:(0.98)> lift:(1.11) lev:(0.05) conv:(4.94)
 3. [ShadeCheck_Good=yes, Pre_Treat_Absent=yes]: 294 ==> [PH_Res_abnorm=yes]: 288    <conf:(0.98)> lift:(1.1) lev:(0.05) conv:(4.76)
 4. [CT_highcont=yes, PH_Neu_normal=yes]: 287 ==> [ShadeCheck_Good=yes]: 281    <conf:(0.98)> lift:(1.16) lev:(0.06) conv:(6.29)
 5. [PH_Neu_normal=yes]: 387 ==> [ShadeCheck_Good=yes]: 376    <conf:(0.97)> lift:(1.15) lev:(0.08) conv:(4.95)
 6. [CM_ReactiveDyes=yes]: 339 ==> [ShadeCheck_Good=yes]: 329    <conf:(0.97)> lift:(1.15) lev:(0.07) conv:(4.73)
 7. [PH_Res_abnorm=yes, PH_Neu_normal=yes]: 348 ==> [ShadeCheck_Good=yes]: 337    <conf:(0.97)> lift:(1.14) lev:(0.07) conv:(4.45)
 8. [PHTest_yes=yes, PH_Neu_normal=yes]: 316 ==> [ShadeCheck_Good=yes]: 306    <conf:(0.97)> lift:(1.14) lev:(0.06) conv:(4.4)
 9. [PH_Res_abnorm=yes, CM_ReactiveDyes=yes]: 307 ==> [ShadeCheck_Good=yes]: 297    <conf:(0.97)> lift:(1.14) lev:(0.06) conv:(4.28)
10. [PH_Res_abnorm=yes, PHTest_yes=yes, PH_Neu_normal=yes]: 282 ==> [ShadeCheck_Good=yes]: 272    <conf:(0.96)> lift:(1.14) lev:(0.06) conv:(3.93)
```

Fig.4 Process Models for the whole log using LRM Association Rule mining algorithm

```
LinkRuleMiner found 12 rules (displaying top 10)

 1. [Pre_Treat_Absent=yes]: 105 ==> [C_Iden_same-1st-two-yrs=yes]: 105    <conf:(1)> lift:(1) lev:(0) conv:(0)
 2. [C_Iden_same-1st-two-yrs=yes]: 105 ==> [Pre_Treat_Absent=yes]: 105    <conf:(1)> lift:(1) lev:(0) conv:(0)
 3. [PH_Res_abnorm=yes]: 102 ==> [Pre_Treat_Absent=yes]: 102    <conf:(1)> lift:(1) lev:(0) conv:(0)
 4. [PH_Res_abnorm=yes]: 102 ==> [C_Iden_same-1st-two-yrs=yes]: 102    <conf:(1)> lift:(1) lev:(0) conv:(0)
 5. [PH_Res_abnorm=yes]: 102 ==> [Pre_Treat_Absent=yes, C_Iden_same-1st-two-yrs=yes]: 102    <conf:(1)> lift:(1) lev:(0) conv:(0)
 6. [Pre_Treat_Absent=yes, PH_Res_abnorm=yes]: 102 ==> [C_Iden_same-1st-two-yrs=yes]: 102    <conf:(1)> lift:(1) lev:(0) conv:(0)
 7. [C_Iden_same-1st-two-yrs=yes, PH_Res_abnorm=yes]: 102 ==> [Pre_Treat_Absent=yes]: 102    <conf:(1)> lift:(1) lev:(0) conv:(0)
 8. [Pre_Treat_Absent=yes]: 105 ==> [PH_Res_abnorm=yes]: 102    <conf:(0.97)> lift:(1) lev:(0) conv:(0.75)
 9. [C_Iden_same-1st-two-yrs=yes]: 105 ==> [PH_Res_abnorm=yes]: 102    <conf:(0.97)> lift:(1) lev:(0) conv:(0.75)
10. [Pre_Treat_Absent=yes]: 105 ==> [C_Iden_same-1st-two-yrs=yes, PH_Res_abnorm=yes]: 102    <conf:(0.97)> lift:(1) lev:(0) conv:(0.75)
```

Fig.5 Process Models for the cluster R0 log using LRM Association Rule mining algorithm

```
Apriori
=======
Best rules found:

 1. C_Iden_same-1st-two-yrs=yes 105 ==> Pre_Treat_Absent=yes 105    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 2. Pre_Treat_Absent=yes 105 ==> C_Iden_same-1st-two-yrs=yes 105    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 3. PH_Res_abnorm=yes 102 ==> Pre_Treat_Absent=yes 102    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 4. PH_Res_abnorm=yes 102 ==> C_Iden_same-1st-two-yrs=yes 102    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 5. PH_Res_abnorm=yes C_Iden_same-1st-two-yrs=yes 102 ==> Pre_Treat_Absent=yes 102    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 6. PH_Res_abnorm=yes Pre_Treat_Absent=yes 102 ==> C_Iden_same-1st-two-yrs=yes 102    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 7. PH_Res_abnorm=yes 102 ==> Pre_Treat_Absent=yes C_Iden_same-1st-two-yrs=yes 102    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 8. Pre_Treat_Absent=yes 105 ==> PH_Res_abnorm=yes 102    <conf:(0.97)> lift:(1) lev:(0) [0] conv:(0.75)
 9. C_Iden_same-1st-two-yrs=yes 105 ==> PH_Res_abnorm=yes 102    <conf:(0.97)> lift:(1) lev:(0) [0] conv:(0.75)
10. Pre_Treat_Absent=yes C_Iden_same-1st-two-yrs=yes 105 ==> PH_Res_abnorm=yes 102    <conf:(0.97)> lift:(1) lev:(0) [0] conv:(0.75)
```

Fig.6 Association rule process model using Apriori algorithm for the cluster R0

```
FPGrowth found 12 rules (displaying top 10)

 1. [Pre_Treat_Absent=yes]: 105 ==> [C_Iden_same-1st-two-yrs=yes]: 105    <conf:(1)> lift:(1) lev:(0) conv:(0)
 2. [C_Iden_same-1st-two-yrs=yes]: 105 ==> [Pre_Treat_Absent=yes]: 105    <conf:(1)> lift:(1) lev:(0) conv:(0)
 3. [PH_Res_abnorm=yes]: 102 ==> [Pre_Treat_Absent=yes]: 102    <conf:(1)> lift:(1) lev:(0) conv:(0)
 4. [PH_Res_abnorm=yes]: 102 ==> [C_Iden_same-1st-two-yrs=yes]: 102    <conf:(1)> lift:(1) lev:(0) conv:(0)
 5. [PH_Res_abnorm=yes]: 102 ==> [Pre_Treat_Absent=yes, C_Iden_same-1st-two-yrs=yes]: 102    <conf:(1)> lift:(1) lev:(0) conv:(0)
 6. [Pre_Treat_Absent=yes, PH_Res_abnorm=yes]: 102 ==> [C_Iden_same-1st-two-yrs=yes]: 102    <conf:(1)> lift:(1) lev:(0) conv:(0)
 7. [C_Iden_same-1st-two-yrs=yes, PH_Res_abnorm=yes]: 102 ==> [Pre_Treat_Absent=yes]: 102    <conf:(1)> lift:(1) lev:(0) conv:(0)
 8. [Pre_Treat_Absent=yes]: 105 ==> [PH_Res_abnorm=yes]: 102    <conf:(0.97)> lift:(1) lev:(0) conv:(0.75)
 9. [C_Iden_same-1st-two-yrs=yes]: 105 ==> [PH_Res_abnorm=yes]: 102    <conf:(0.97)> lift:(1) lev:(0) conv:(0.75)
10. [Pre_Treat_Absent=yes]: 105 ==> [C_Iden_same-1st-two-yrs=yes, PH_Res_abnorm=yes]: 102    <conf:(0.97)> lift:(1) lev:(0) conv:(0.75)
```

Fig.7 Association rule process model using FPGrowth algorithm for the cluster R0

```
HMine found 12 rules (displaying top 10)

 1. [Pre_Treat_Absent=yes]: 105 ==> [C_Iden_same-1st-two-yrs=yes]: 105   <conf:(1)> lift:(1) lev:(0) conv:(0)
 2. [C_Iden_same-1st-two-yrs=yes]: 105 ==> [Pre_Treat_Absent=yes]: 105   <conf:(1)> lift:(1) lev:(0) conv:(0)
 3. [PH_Res_abnorm=yes]: 102 ==> [Pre_Treat_Absent=yes]: 102   <conf:(1)> lift:(1) lev:(0) conv:(0)
 4. [PH_Res_abnorm=yes]: 102 ==> [C_Iden_same-1st-two-yrs=yes]: 102   <conf:(1)> lift:(1) lev:(0) conv:(0)
 5. [PH_Res_abnorm=yes]: 102 ==> [Pre_Treat_Absent=yes, C_Iden_same-1st-two-yrs=yes]: 102   <conf:(1)> lift:(1) lev:(0) conv:(0)
 6. [Pre_Treat_Absent=yes, PH_Res_abnorm=yes]: 102 ==> [C_Iden_same-1st-two-yrs=yes]: 102   <conf:(1)> lift:(1) lev:(0) conv:(0)
 7. [C_Iden_same-1st-two-yrs=yes, PH_Res_abnorm=yes]: 102 ==> [Pre_Treat_Absent=yes]: 102   <conf:(1)> lift:(1) lev:(0) conv:(0)
 8. [Pre_Treat_Absent=yes]: 105 ==> [PH_Res_abnorm=yes]: 102   <conf:(0.97)> lift:(1) lev:(0) conv:(0.75)
 9. [C_Iden_same-1st-two-yrs=yes]: 105 ==> [PH_Res_abnorm=yes]: 102   <conf:(0.97)> lift:(1) lev:(0) conv:(0.75)
10. [Pre_Treat_Absent=yes]: 105 ==> [C_Iden_same-1st-two-yrs=yes, PH_Res_abnorm=yes]: 102   <conf:(0.97)> lift:(1) lev:(0) conv:(0.75)
```

Fig.8 Association rules process model using H-Mine algorithm for the cluster R0

```
LinkRuleMiner found 12 rules (displaying top 10)

 1. [Pre_Treat_Absent=yes]: 105 ==> [C_Iden_same-1st-two-yrs=yes]: 105   <conf:(1)> lift:(1) lev:(0) conv:(0)
 2. [C_Iden_same-1st-two-yrs=yes]: 105 ==> [Pre_Treat_Absent=yes]: 105   <conf:(1)> lift:(1) lev:(0) conv:(0)
 3. [PH_Res_abnorm=yes]: 102 ==> [Pre_Treat_Absent=yes]: 102   <conf:(1)> lift:(1) lev:(0) conv:(0)
 4. [PH_Res_abnorm=yes]: 102 ==> [C_Iden_same-1st-two-yrs=yes]: 102   <conf:(1)> lift:(1) lev:(0) conv:(0)
 5. [PH_Res_abnorm=yes]: 102 ==> [Pre_Treat_Absent=yes, C_Iden_same-1st-two-yrs=yes]: 102   <conf:(1)> lift:(1) lev:(0) conv:(0)
 6. [Pre_Treat_Absent=yes, PH_Res_abnorm=yes]: 102 ==> [C_Iden_same-1st-two-yrs=yes]: 102   <conf:(1)> lift:(1) lev:(0) conv:(0)
 7. [C_Iden_same-1st-two-yrs=yes, PH_Res_abnorm=yes]: 102 ==> [Pre_Treat_Absent=yes]: 102   <conf:(1)> lift:(1) lev:(0) conv:(0)
 8. [Pre_Treat_Absent=yes]: 105 ==> [PH_Res_abnorm=yes]: 102   <conf:(0.97)> lift:(1) lev:(0) conv:(0.75)
 9. [C_Iden_same-1st-two-yrs=yes]: 105 ==> [PH_Res_abnorm=yes]: 102   <conf:(0.97)> lift:(1) lev:(0) conv:(0.75)
10. [Pre_Treat_Absent=yes]: 105 ==> [C_Iden_same-1st-two-yrs=yes, PH_Res_abnorm=yes]: 102   <conf:(0.97)> lift:(1) lev:(0) conv:(0.75)
```

Fig.9 Association rule process model using LinkRuleMiner algorithm for the cluster R0

REFERENCES

[1] H. Mannila, H. Toivonen, and A.I. Verkamo. “Discovery of Frequent Episodes in Event Sequences. Data Mining and Knowledge Discovery”, 1997, Vol. 1, Issue. 3, pp.259–289.

[2] Margaret. C Perivoliotis, Wax Resist Decoration, “An Ancient Mediterranean Art”, Published by on line journal antciencia.com, ISSN 1646-3463, Aug-Oct’ 2000, Vol. 2, No. 4.

[3] G. Greco, A. Guzzo, and L. Pontieri, “Discovering Expressive Process Models by Clustering Log Traces”, IEEE Transactions on Knowledge and Data Engineering, Aug’ 2006, Vol. 18, Issue. 8, pp. 1010-1027.

[4] B. Mirkin, *Clustering for data mining: a data recovery approach*. Publisher London: Chapman and Hall/CRC, 2005.

[5] El Mogahzy, Y.E., “Selecting Cotton Fiber Properties for Fitting Reliable Equations to HVI Data”. Textile Research Journal, 1988, Vol. 58, No. 7, pp. 392-397.

[6] Agrawal, R., Imielinski, T., and Swami, A. N, “Mining association rules between sets of items in large databases”, In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, 1993, pp. 207-216.

[7] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules”, In Proc. of the 20th International Conference on Very Large Databases, Santiago, Chile, September 1994.

[8] Bowen, P.L., O'Farrell, R.A., and Rohde. F, "Analysis of Competing Data Structures: Does Ontological Clarity Produce Better End User Query Performance", Journal of the Association for Information Systems, 2006, Vol. 7, Issue 8, pp. 514-544.