Journal of Theoretical and Applied Information Technology

www.jatit.org

DERIVING CLUSTER KNOWLEDGE USING ROUGH SET THEORY

[1]Shuchita Upadhyaya , [2]Alka Arora, [3] Rajni Jain

[1] Deptt. of Computer Science and Application, Kurukshetra University, Kurukshetra, India.

[2] Indian Agricultural Statistics Research Institute, Library Avenue, Pusa, New Delhi-110012, India

[3] National Center for Agricultural Economics and Policy Research, Library Avenue, Pusa, New Delhi-110012, India

Email: [1]shuchita_bhasin@yahoo.com, [2]alkak@iasri.res.in, [3]rajni@ncap.res.in

ABSTRACT

Clustering algorithms gives general description of the clusters listing number of clusters and member entities in those clusters. It lacks in generating cluster description in the form of pattern. Deriving pattern from clusters along with grouping of data into clusters is important from data mining perspective. In the proposed approach reduct from rough set theory is employed to generate pattern. Reduct is defined as the set of attributes which distinguishes the entities in a homogenous cluster. It is observed that most of the remaining attributes in the cluster has same value for their attribute value pair. Reduct attributes are removed to formulate pattern by concatenating most contributing attributes. Proposed approach is demonstrated using benchmarking mushroom dataset from UCI repository.

Keywords: *Rough set theory, Reduct, Indiscernibility, Clustering, Pattern discovery, Cluster description, Mushroom*

1 INTRODUCTION

In data mining, clustering is used as a tool for finding patterns and regularities within the data. Clustering algorithms groups the entities into different clusters such that entities in a cluster are highly similar and hence the entities belonging to different clusters are highly dissimilar. Pattern is defined as a logical statement describing the cluster structure in terms of relevant attributes. Clustering algorithms in literature are divided into different categories. Partitioning clustering algorithms are commonly used clustering algorithms [5]. Partitioning algorithms divides the data into k non overlapping clusters, where k is the number of clusters specified by the user as input. These clustering algorithms, only generates general description of the clusters depicting member entities of each cluster. However, it lacks in generating pattern as this approach has no mechanism for selecting and evaluating the attributes in the process of generating clusters [7]. Post processing of clusters is required in data mining for deriving useful knowledge in the form of pattern. Pattern is formulated by conjunction of significant attribute value pair and hence describes the cluster in more meaningful format. Producing a pattern is of interest in the situation where there is a need to study the relationship describing the data. It is also useful in a situation where interpretation of clusters is required in user understandable format.

In this paper, an attempt is being made using Rough Set Theory (RST) to derive patterns from the clusters obtained using partitioning clustering algorithm. RST divides the data into indiscernible classes. RST has a natural appeal to be applied in clustering as these indiscernible classes can be construed as clusters. Moreover, RST also performs automatic concept approximation by producing minimal subset of attributes (Reduct) that can distinguish all the entities in the dataset. Our aim is to generate pattern of individual clusters and hence reduct is computed for each cluster. If reducts are removed from the cluster, remaining attributes in the cluster will have same attribute value pair. These remaining attributes play significant role in pattern generation. Pattern is then formulated with the conjunction of major contributing attributes. The efficacy of the approach is demonstrated with the help of benchmarking mushroom dataset from the UCI repository [13]. Objective of applying the proposed approach on mushroom dataset is to study the relationship of attributes with edible and poisonous nature of mushrooms.

The paper is organized as follows: In section 2 the basic notions of rough set theory and cluster

description is described. Section 3 presents the cluster description approaches including the proposed approach. In section 4 application of the proposed approach is demonstrated on mushroom dataset followed by conclusion in Section 5.

2 BASIC NOTIONS

2.1. Rough Set Theory Concepts

In RST, data is represented by an information system $X = (U, A \cup \{d\})$ [6, 11]. In this U is non-empty finite set of entities and A is a non-empty, finite set of attributes on U, where $d \notin A$ is decision/class attribute. With every attribute $a \in A$, we associate a set V_a such that $a : U \to V_a$. The set V_a is called the domain or value set of attribute a. Every entity x, in an information system X, is characterized by its information vector:

$$InfX(x) = \{(a, a(x)) : a \in A\}$$

Relationship between entities is described by their attribute values. Indiscernibility relation $IND(B)$, for any subset $B \subseteq A$ is defined by:

$$x \; IND(B) \; y \Leftrightarrow \forall_{a \in B} (a(x) = a(y))$$

Two entities are considered to be indiscernible by the attributes in B, if and only if they have the same value for every attribute in B. Entities in the information system about which we have the same knowledge form an equivalence relation. $IND(B)$ is an equivalence relation that partitions U into equivalence classes. Set of such partitions are denoted by $U / IND(B)$.

Reduct is the set of attributes that can differentiate all equivalence classes. Mostly reduct is computed relative to decision attribute in the dataset. However, our approach of reduct computation is different. Clustering is done on unsupervised data where decision/class information is not present and hence reduct computation is purely on the basis of indiscernibility. We have computed reduct for individual cluster as compared to reduct computation for dataset because our aim is to generate patterns of individual clusters. There are various methods and software's available for computation of reducts. We have used genetic algorithm for reduct computation using Rosetta software.

2.2. Illustration

Small table from mushroom dataset is considered for illustration of RST concepts (Table 1).

Table 1: Small mushroom data set

Id	cap-shape	cap-surface	cap-color	bruises	odor
X1	b	s	b	t	n
X2	f	s	w	t	n
X3	f	s	p	t	n
X4	b	s	b	t	n
X5	b	s	w	t	n
X6	b	y	w	t	n
X7	f	y	p	t	n
X8	b	s	b	t	n
X9	b	y	p	t	n

Using Table 1 some concepts of RST described in section 2.1 are:

U={X1, X2, X3, X4, X5, X6, X7, X8, X9}

A= {cap-shape, cap-surface, cap-color, bruises, odor}

Vcap-shape = {b, f}, Vcap-surface ={s, y}, Vcap-color={b, p, w}, Vbruises={t}, Vodor={n},

InfX(X1)= {cap-shape, b}, i.e value of attribute cap-shape for entity X1is b.

For any subset $B \subseteq A$, when B= {cap-shape}then entities {X1, X4, X5, X6, X8} (X2,X3,X7) in these sets are indiscernible and form different equivalence classes. Therefore U/IND(B)={(X1,X4,X5,X6,X8),(X2,X3,X7)}

Similarly for B={cap-surface, odor}:

U/IND(B)={(X1,X2,X3,X4,X5,X8), (X6,X7,X9)}

Computation of reduct using genetic algorithm resulted in reduct set (R) of attributes: R={cap-shape, cap-surface, cap-color}.

2.3. Cluster Description concepts

In the information system, attribute value pair of the form $D = (a \in V_a)$ is defined as descriptor and the value set V_a is called the range of D [10]. Support for the descriptor D is defined as the number of entities from U satisfying D. To measure and compare the describing capabilities of various descriptors, Precision Error (PE) of the descriptor is computed. PE of descriptor D is defined as:

$$PE(D) = \frac{\left| \; false \; positive \; C(D) \right|}{\left| U - C \right|}$$

where numerator defines the number of entities that lies outside cluster C for which descriptor ($D = a$) is true and denominator defines the number of entities outside cluster C. A descriptor is said to be more contributing if it has less PE which essentially means that most of the entities satisfying that descriptor belongs to a single cluster.

Now cluster description can be defined in terms of pattern $P = \bigwedge_{i=1}^{n} D_i \quad \forall C$ that is formed by the conjunction of most contributing descriptors from that cluster. Similarly PE of pattern P for cluster C is defined as [8]:

$$PE\ (P) = \frac{\left| \textit{false positive}\, C(P) \right|}{\left| U - C \right|}$$

where numerator denotes the number of entities that lies outside the cluster C for which pattern P is true and denominator denotes the number of entities outside cluster C.

Therefore problem of pattern formulation can be carried out by combining the attributes with less PE such that PE for P equals zero, means this pattern P distinctively describes the cluster with no errors.

3 CLUSTER DESCRIPTION APPROACHES

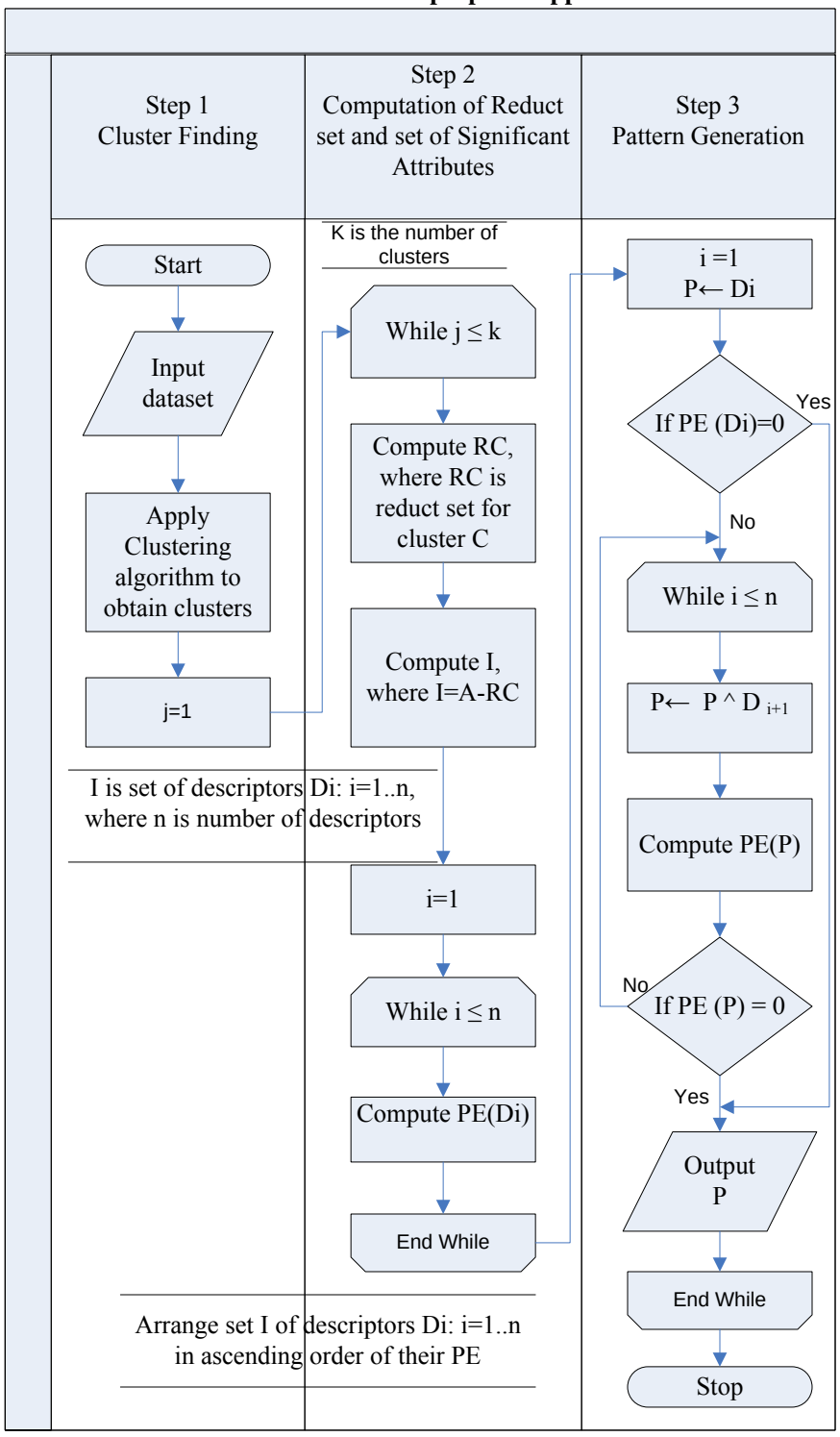
3.1. Review of Literature

The field of producing patterns for individual clusters is relatively new. There are few references of cluster description approaches available in literature. Mirkin has proposed a method for cluster description applicable to only continuous attributes [8]. In Mirkin's approach attributes are normalized first and then ordered according to their contribution weights which are proportional to the squared differences between their with-in group averages and grand means. A conjunctive description of cluster is then formed by consecutively adding attributes according to the sorted order. An attribute is added to the description only if it decreases the error. This forward attribute selection process stops after the last element of attribute set is checked. Abidi et al. has proposed the rough set theory based method for rule creation for unsupervised data using dynamic reduct [1, 2]. Dynamic reduct is defined as the frequently occurring reduct in the population of reduct set obtained using genetic algorithm. However these approaches have its limitations. Mirkin's approach is applicable only to datasets having continuous attributes. Abidi in his approach has used the cluster information obtained after cluster finding and generated rules from entire data with respect to decision attribute, instead of producing description for individual clusters. Other popular pattern generation approach like decision tree [10] is not directly applicable to clustering as criteria in clustering is to get homogenous clusters with respect to all the attributes. However in decision tree homogeneity is with respect to decision attribute.

3.2. Proposed Approach

Proposed approach of pattern formulation is divided into three parts. First part deals with obtaining clusters from dataset by applying clustering algorithm. In the second stage we have computed sets of significant and non significant attributes for that cluster. As cluster is set of similar data entities, only similar attribute value pair are significant for that cluster and rest are non significant. Computation of reduct set (RC) in a cluster will provide the set of non significant attributes for that cluster, as reduct accounts for discerning between the entities. These non significant attributes (reduct) can be straight away removed from the cluster. The remaining attributes now termed as descriptors (ref section 2.3) form the set of significant attributes (I) for that cluster. Contributions of these descriptors in a cluster are measured in terms of PE. PE is calculated for every descriptor (D) in set I, and descriptors are arranged in ascending order of their PE. In the third stage pattern is formulated by conjunction of descriptors with less PE such that PE for the pattern equals zero.

Flowchart of proposed approach:



4 EXPERIMENTAL RESULTS

4.1. Data description

We have considered benchmarking mushroom dataset from UCI repository for demonstration of proposed approach [14]. Dataset consists of large number of records that is 8124 records. The number of edible and poisonous mushrooms in the data set is 4208 and 3916 respectively. Table 1 shows the 22 categorical attributes that describes the physical characteristics of mushrooms. Class attribute (edible (e) or poisonous (p)) and attribute stalk root with missing values are not considered for clustering.

Table 1: Attribute Information of Mushroom dataset

1.	cap-shape: bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s
2.	cap-surface: fibrous=f, grooves=g, scaly=y, smooth=s
3.	cap-color: brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y
4.	bruises: bruises=t, no=f
5.	odor: almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s
6.	gill-attachment: attached=a, descending=d, free=f, notched=n
7.	gill-spacing: close=c, crowded=w, distant=d
8.	gill-size: broad=b, narrow=n
9.	gill-color: black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y
10.	stalk-shape: enlarging=e, tapering=t
11.	stalk-root: bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?
12.	stalk-surface-above-ring: ibrous=f, scaly=y, silky=k, smooth=s
13.	stalk-surface-below-ring: ibrous=f, scaly=y, silky=k, smooth=s
14.	stalk-color-above-ring: brown=n ,buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
15.	stalk-color-below-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
16.	veil-type: partial=p, universal=u
17.	veil-color: brown=n, orange=o, white=w, yellow=y
18.	ring-number: none=n, one=o, two=t
19.	ring-type: cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z
20.	spore-print-color: black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y
21.	population: abundant=a, clustered=c, numerous=n, scattered=s, several=v, solitary=y
22.	habitat: grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d

4.2. Data Clustering

We have used Weka implementation [15] of EM algorithm for cluster finding. EM is a mixture based algorithm that attempts to maximize the likelihood of the model [9]. EM models the distribution of the entities probabilistically, so that an entity belongs to a cluster with certain probability. The first step, calculation of the cluster probabilities, which are the expected class value, is "expectation"; the second step is calculation of the distribution parameter is "maximization" of the likelihood of the distribution given the data. By default, EM selects the number of clusters automatically by maximizing the logarithm of the likelihood of future data, estimated using cross-validation. Beginning with one cluster, it continues to add clusters until the estimated log-likelihood decreases.

When EM clustering algorithm is applied on mushroom dataset, it learned 14 numbers of clusters from the data. Table 2 shows the result obtained with EM algorithm. There is wide variance among the size of the clusters that range from 96 entities to 1728 entities. As shown in Table 2, except clusters

3, 11 and 13 which are mix clusters, all other clusters are pure clusters. Pure clusters in the sense that mushrooms in every cluster are either all poisonous or all edible.

Table 2: Clustering results with EM algorithm

Cluster Number	poisonous(p)	edible(e)
1	288	0
2	1728	0
3	84	112
4	0	192
5	0	768
6	0	96
7	0	1728
8	256	0
9	1296	0
10	1	511
11	192	96
12	0	192
13	72	224
14	0	288

4.3. Reduct Computation

We have used Genetic Algorithm (GA) based method [15] for reduct computation for our experiments in this paper using Rosetta software [12]. To study the characteristics of poisonous mushrooms, reduct analysis is carried out on individual pure poisonous clusters (1, 2, 8, and 9) without considering the decision information. Table 3 shows the reduct attributes in pure poisonous clusters.

Table 3: Reduct attribute in poisonous clusters

Cluster1	cap-shape, cap-color, gill-color, stalk-surface-above-ring, stalk-surface-below-ring, population, habitat
Cluster2	cap-shape, cap-surface, cap-color, odor, stalk-surface-above-ring, stalk-surface-below-ring, stalk-color-above-ring, stalk-color-below-ring, habitat
Cluster8	cap-shape, cap-surface, cap-color, gill-color, spore-print-color, population, habitat
Cluster9	cap-shape, cap-surface, cap-color, gill-color, stalk-color-above-ring, stalk-color-below-ring, population, habitat

Although all the four clusters are poisonous, yet reduct attributes are not common among these clusters. Some attributes may be playing role in one cluster and may not significant in other clusters. Similarly to study the characteristics of edible mushrooms, reduct analysis is carried out on individual pure edible clusters (4, 5, 6, 7, 10, 12 and 14). Table 4 shows the reduct attributes in pure edible clusters.

Table 4: Reduct attributes in edible clusters

Cluster4	cap-shape, gill-color, veil-color, spore-print-color, population
Cluster5	cap-shape, cap-surface, cap-color, gill-color, stalk-surface-above-ring, stalk-surface-below-ring, spore-print-color, population
Cluster6	cap-shape, cap-surface, cap-color, odor, gill-color, spore-print-color
Cluster7	cap-shape, cap-surface, cap-color, gill-color, stalk-color-above-ring, stalk-color-below-ring, spore-print-color, population
Cluster10	cap-shape, cap-surface, cap-color, odor, gill-color, spore-print-color, population, habitat
Cluster12	cap-shape, cap-color, odor, gill-color, spore-print-color, population, habitat
Cluster14	cap-shape, cap-surface, cap-color, gill-color, stalk-surface-above-ring, stalk-surface-below-ring, population

4.4. Cluster Description

Let us consider Cluster1 for pattern generation. As a cluster is defined as set of similar data entities and reduct attributes accounts for discerning entities with in cluster, therefore these can be clear cut removed from the cluster. When we remove the reduct attributes (Table 3) of cluster1, remaining descriptors in cluster1 are cap-surface=s, bruises=t, odor=f, gill-attachment=f, gill-spacing=c, gill-size=b, stalk-shape=t, stalk-color-above-ring=w, stalk-color-below-ring=w, veil-color=w, ring-number=o, ring-type=p, spore-print-color=h. All these descriptors have same value for all the entities within this cluster, therefore these are the contributing descriptors for this cluster. We then calculated PE for these descriptors to find out the major contributing descriptors. Let us consider calculation of PE for cap-surface=s and bruises=t in cluster1 which contains 288 entities. Descriptors cap-surface=s has support of 2556 entities and bruises=t has support of 3376 entities in the dataset. PE is defined as number of false positive for that descriptor divided by the total number of entities outside that cluster.

Therefore PE (cap-surface=s) = (2556-288)/ (8124-288) = .2894

Similarly PE (bruises=t) = (3376-288)/ (8124-288) =.3940.

In the similar way PE is computed for descriptors of every cluster. Table 5 and Table 6 show the descriptors along with value of PE for pure poisonous and edible clusters respectively.

Table 5: PE for descriptors in poisonous clusters

Cluster 1	spore-print-color=h(.1715), odor=f(.2390), cap-surface=s(.2894), bruises=t(.3940), ring-type=p(.4696), stalk-color-below-ring=w(.5227), stalk-color-above-ring=w(.5329), stalk-shape=t(.5513), gill-size=b(.6794), gill-spacing=c(.8325), ring-number=o(.9188), gill-attachment=f(.9732), veil-color=w(.9744)
Cluster 2	gill-color=b(0), spore-print-color=w(.1031), gill-size=n(.1225), ring-type=e(.1638), population=v(.3614), stalk-shape=t(.4502), bruises=f(.4721), gill-spacing=c(.7948), ring-number=o(.9005), gill-attachment=f(.9671), veil-color=w(.9687)
Cluster 8	odor=p(0), gill-size=n(.2867), bruises=t(.3965), stalk-shape=e(.4143), ring-type=p(.4717) stalk-color-below-ring=w(.5246), stalk-color-above-ring=w(.5348), stalk-surface-below-ring=s(.5948), stalk-surface-above-ring=s(.6253), gill-spacing=c(.8332), ring-number=o(.9191), veil-color=w(.9745), gill-attachment=f(.9733)
Cluster 9	ring-type=l(0), spore-print-color=h(.0492), odor=f(.1266), stalk-surface-below-ring=k(.1476), stalk-surface-above-ring=k(.1575), stalk-shape=e(.3251), bruises=f(.5055), gill-size=b(.6321), gill-spacing=c(.8078), ring-number=o(.9068), gill-attachment=f(.9692), veil-color=w(.9707)

Table 6: PE for descriptors in edible clusters

Cluster4 (192 entities)	stalk-color-above-ring=o(0), stalk-color-below-ring=o(0), gill-attachment=a(.0022), habitat=l(.0806), cap-color=n(.2637), cap-surface=s(.2980), stalk-shape=e(.4190), odor=n(.4205), ring-type=p(.4760), bruises=f(.5743), stalk-surface-below-ring=s(.5980), stalk-surface-above-ring=s(.6283), gill-size=b(.6833), gill-spacing=c(.8345), ring-number=o(.9198)
Cluster5 (768 entities)	gill-spacing=w(.0739), habitat=g(.1876), ring-type=e(.2729), odor=n(.3752), stalk-color-below-ring=w(.4915), stalk-color-above-ring=w(.5024), stalk-shape=t(.5220), bruises=f(.5410), gill-size=b(.6585), ring-number=o(.9135), gill-attachment=f(.9714), veil-color=w(.9728)
Cluster6 (96 entities)	gill-spacing=w(.1514), habitat=d(.3801), gill-size=n(.3009), bruises=t(.4085), ring-type=p(.4823), population=v(.4912), stalk-color-below-ring=w(.5341), stalk-color-above-ring=w(.5440), stalk-shape=t(.5620), stalk-surface-below-ring=s(.6028), stalk-surface-above-ring=s(.6327), veil-color=w(.9750), ring-number=o(.9207), gill-attachment=f(.9738).
Cluster7 (1728 entities)	habitat=d(.2220), bruises=t(.2576), odor=n(.2814), ring-type=p(.3502), stalk-shape=t(.4502), gill-attachment=f(.9671), stalk-surface-below-ring=s(.5015), stalk-surface-above-ring=s(.5390), gill-size=b(.6072), gill-spacing=c(.7948), ring-number=o(.9005), veil-color=w(.9687)
Cluster10 (511 entities)	bruises=t(.3763), stalk-shape=e(.3947), ring-type=p(.4540), stalk-color-below-ring=w(.5087), stalk-color-above-ring=w(.5192), stalk-surface-below-ring=s(.5812), stalk-surface-above-ring=s(.6127), gill-size=b(.6700), gill-spacing=c(.8276), ring-number=o(.9164), gill-attachment=f(.9724), veil-color=w(.9737)
Cluster12 (192 entities)	stalk-surface-below-ring=y(.0115), cap-surface=y(.3847), bruises=t(.4014), stalk-shape=e(.4190), ring-type=p(.4760), stalk-color-below-ring=w(.5284), stalk-color-above-ring=w(.5385), stalk-surface-above-ring=s(.6283), gill-size=b(.6833), gill-spacing=c(.8345), ring-number=o(.9198), gill-attachment=f(.9735), veil-color=w(.9747)
Cluster14 (288 entities)	ring-number=t(.0398), gill-spacing=w(.1306), habitat=g(.2373), spore-print-color=w(.2679), stalk-shape=e(.4119), odor=n(.4134), ring-type=p(.4696), stalk-color-below-ring=w(.5227), stalk-color-above-ring=w(.5329), bruises=f(.5691), gill-size=b(.6794), gill-attachment=f(.9732), veil-color=w(.9744)

For pattern generation we look for descriptors with zero or less PE. If PE for any descriptor in the cluster is not equal to zero then pattern is formed with conjunction of descriptors with less PE such that PE for pattern equals zero. For example pattern generation for Cluster1 involves conjunction of descriptors spore-print-color=h with odor=f and cap-surface=s such that this pattern describes the Cluster1 with no errors. Similarly, in Cluster2, descriptor gill-color=b has zero PE therefore this alone describes the cluster with no errors.

4.5. Results

Cluster description with proposed approach resulted in following patterns for the poisonous clusters.

Cluster1 (288 entities): spore-print-color=h ^ odor=f ^ cap-surface=s.

Cluster2 (1728 entities): gill-color=b.

Cluster8 (256 entities): odor=p.

Cluster9 (1296 entities): ring-type=l

Pattern obtained with proposed approach for edible clusters are:

Cluster4 (192 entities): stalk-color-above-ring=o or stalk-color-below-ring=o.

Cluster5 (768 entities): gill-spacing =w ^ habitat=g ^ ring-type=e

Cluster6 (96 entities): gill-spacing=w ^ gill-size=n ^ habitat=d ^ bruises=t.

Cluster7 (1728 entities): habitat=d ^ bruises=t ^ odor=n.

Cluster10 (511 entities): bruises=t ^ stalk-shape=e ^ ring-type=p ^ stalk-surface-below-ring=y ^ gill-size=b ^ ring-number=o.

Cluster12 (192 entities): stalk-surface-below-ring=y^ cap-surface=y ^ bruises=t.

Cluster14 (288 entities): ring-number=t ^ gill-spacing=w.

5 CONCLUSION

Reduct driven approach for pattern generation from clusters is presented in this paper on benchmarking dataset. Reduct is computed for individual clusters for filtering non-significant attributes. Precision error is then computed on remaining significant attributes. Pattern is then formulated from most contributing attributes. It is observed that patterns obtained with this approach, distinctively described the clusters with no errors. To confirm the existence of relation, future research will be focused on applying the same approach on more benchmarking datasets.

6 REFERENCES

[1] Abidi, S. S. R., Hoe, K. M., Goh, A, "Analyzing data clusters: A rough set approach to extract cluster defining symbolic rules", in *Fisher, H., Hoffman, A. (eds.) Lecture notes in Computer Science: Advances in Intelligent Data Analysis, 4th Intl. Symposium, IDA-01*, Springer Verlag, Berlin, 2001.

[2] Abidi, S. S. R., Goh, A., "Applying knowledge discovery to predict infectious disease epidemics", in *Lee, H., Motoda, H. (eds.) Lecture notes in artificial intelligence, PRICAI'98:* Topics in artificial intelligence, Springer Verlag, Berlin, 1998.

[3] Ganter, B., Wille, R., *Formal Concept Analysis: Mathematical Foundations*, Springer- Verlag, New York Inc., 1997.

[4] Han, J., Kamber, M., *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2001.

[5] Jain, A.K., Murty, M.N., Flynn, P.J., "Data Clustering: A review", *ACM Computing Surveys*, Vol. 31 N. 3, pp. 264-323, 2001.

[6] Komorowski, J., Pawlak, Z., Polkowski, S., "Rough sets: A tutorial", in *Pal, S.K., Skowron, A. (eds.) Rough Fuzzy Hybridization: A new Trend in Decision-Making,* Springer-Verlag Berlin, pp. 3–98, 1999.

[7] Michalski, R.S., Stepp, R.E., "Clustering", in *Shapiro, S.C. (eds.) Encyclopedia of artificial intelligence*, J. Wiley & Sons, New York, 1992.

[8] Mirkin, B., "Concept Learning and Feature Selection based on Square-Error Clustering", *Machine Learning*, 35, pp.25-40, 1999.

[9] Mirkin, B., *Clustering for Data Mining: Data Recovery Approach*, Chapman & Hall/CRC, 2005.

[10] Nguyen S. H., Nguyen T. T., Skowron A., Synak P., " Knowledge discovery by Rough Set Methods", in *Proc. of the International Conference on Information Systems Analysis and Synthesis*, Orlando, USA, pp. 26-33, 1996.

[11] Pawlak, Z., *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, 1991.

[12] Rosetta, http://www.rosetta.com

[13] UCI: Repository of machine learning databases, http://www.ics.uci.edu/~mlearn/

[14] WEKA: A Machine Learning Software, http://www.cs.waikato.ac.nz/~ml/

[15] Wroblewski, J, " Finding minimal reducts using genetic algorithms", *in Wang [513]*, pp.186-189, 1995.