MINING OF COMPLEX OBJECTS VIA DESCRIPTION CLUSTERING

Alejandro García López
European Laboratory for Nuclear Research (CERN)
Geneva (Switzerland)

Rafael Berlanga
Depto. de Lenguajes y Sistemas Informáticos. Universitat Jaume I
Castellón (España)

Roxana Danger
Depto. de Lenguajes y Sistemas Informáticos. Universitat Jaume I
Castellón (España)

Keywords: Complex objects, association rules, clustering, data mining.

Abstract: In this work we present a formal framework for mining complex objects, being those characterised by a set of heterogeneous attributes and their corresponding values. First we will do an introduction of the various Data Mining techniques available in the literature to extract association rules. We will as well show some of the drawbacks of these techniques and how our proposed solution is going to tackle them. Then we will show how applying a clustering algorithm as a pre-processing step on the data allow us to find groups of attributes and objects that will provide us with a richer starting point for the Data Mining process. Then we will define the formal framework, its decision functions and its interesting measurement rules, as well as a newly designed Data Mining algorithms specifically tuned for our objectives. We will also show the type of knowledge to be extracted in the form of a set of association rules. Finally we will state our conclusions and propose the future work.

1 INTRODUCTION

The problem of mining complex objects, as we understand it, is that of extracting useful information out of multidimensional heterogeneous data. To fully comprehend this concept we need therefore to define what we mean by *extracting useful information* and *multidimensional heterogeneous data*.

When we talk about *multidimensional heterogeneous data*, we are referring to collections of attributes of different types (boolean, categorical, numerical, etc.) which are represented in an structured way. This structured representation would normally be based on a relational schema, although we could also think of, for example, a collection of XML documents.

On the other hand, what we mean by *extracting useful information* is mainly the discovering of frequent and approximate underlying patterns (Association Rules, ARs), which can help users to undertake a number of decision taking tasks. Examples of these are: summarizing a data collection, finding interesting relations amongst its attributes, finding certain trends, etc.

This kind of association rules can be applied to a wide range of applications. Our main motivating application consists of mining large log repositories that contain data about the performance of a GRID infrastructure for ALICE experiments at CERN. Stored data records include heterogeneous attributes involving different data types (e.g. location of a node, average serving time, number of processes, etc.) In this context, users can be interested on finding frequent patterns amongst these attribtues in order to plan properly the distribution of tasks over the GRID.

The definition of ARs was first stated in (Agrawal et al., 1993), referring to binary attributes. Basically it is defined as follows. Let $I = I_1, I_2, ..., I_m$ be a set of binary attributes, called items. Let T be a database of transactions. Each transaction t is represented as a binary vector, with $t[k] = 1$ if t bought the item I_k, and $t[k] = 0$ otherwise. Let X be a set of some

García López A., Berlanga R. and Danger R. (2006).
MINING OF COMPLEX OBJECTS VIA DESCRIPTION CLUSTERING.
In *Proceedings of the First International Conference on Software and Data Technologies*, pages 187-194
DOI: 10.5220/0001318401870194

items in I. We say that a transaction t satisfies X if for all items I_k in X, $t[k] = 1$. An AR is then, an implication of the form $X \Rightarrow I_j$, where X is a set of some items in I, and I_j is a single item in I that is not present in X. An example of this type of rule is: "90% of transactions that purchased bread and butter also purchased milk". The antecedent of this rule consists of bread and butter and the consequent consists of milk alone.

In (Srikant and Agrawal, 1996) where the concept of Quantitative Association Rules (QARs) is first shown, the authors deal with the fact that the vast majority of relational databases, either based on scientific or business information are not filled with binary datatypes (as requested by the classical ARs) but with a much richer range of datatypes both numerical and categorical.

A first approach to tackle this problem consists of mapping the QARs problem into the *boolean* ARs problem. The key idea is that if all attributes are categorical or the quantitative attributes have only a few values, this mapping is straightforward. However, this approach generates problems as if the intervals are too large, some rules may not have the required *minimum confidence* and if they are too small, some rules may not have the required *minimum support*. We could also think of the strategy of considering all possible continuous ranges over the values of the quantitative attribute to cover the partitioned intervals (to solve the *minimum confidence* problem) and increase the number of intervals (solving the problem of *minimum support*). Unfortunately two new problems arise: First, if a quantitative attribute has *n* values (or intervals), there are on average $O(n^2)$ ranges that include a specific value or interval, fact that blows up the execution time and second, if a value (or interval) of a quantitative attribute has *minimum support*, so will any range containing this value/interval, therefore, the number of rules increases dramatically.

The approach taken by (Srikant and Agrawal, 1996) is different. Considering ranges over adjacent values/intervals of quantitative attributes to avoid the *minimum support* problem. To mitigate the problem of the excess of execution time, they restricted the extent to which adjacent values/intervals may be combined by introducing a user-specified *maximum support* parameter; they stop combining intervals if their combined *support* exceeds this value. They introduce as well a *partial completeness measure* in order to be able to decide whether to partition a quantitative attribute or not and how many partitions should there be, in case it's been decided to partition at all. To address the problem of the appearance of too many rules, they propose an *interest measure* based on the deviation from the expectation that helps to prune out the uninteresting rules (extension of the *interest measure* already proposed in (Srikant and Agrawal, 1997)). Finally an algorithm to extract QARs is presented, sharing the same idea of the algorithm for finding ARs over binary data given in (Agrawal and Srikant, 1994) but adapting the implementation to the computational details of how candidates are generated and how their *supports* are now counted.

In (Miller and Yang, 1997), the authors pointed out the pitfalls of the equi-depth method (interest measure based on deviation), and presented several guiding principles for quantitative attribute partitioning. They apply clustering methods to determine sets of dense values in a single attribute or over a set of attributes that have to be treated as a whole. But although they took distance among data into account, they did not take the relations among other attributes into account by clustering a quantitative attribute or a set of quantitative attributes alone. Based on this, (Tong et al., 2005) improved the method to take into account the relations amongst attributes.

Another improvement in the mining of quantitative data is the inclusion of Fuzzy Sets to solve the *sharp boundary problem* (Kuok et al., 1998). An element belongs to a set category with a membership value, but it can as well belong to the neighbouring ones.

In (Dong and Tjortjis, 2003) a mixed approach based on the *quantitative approach* introduced by (Srikant and Agrawal, 1996), the hash-based technique from the Direct Hashing and Pruning (DHP) algorithm (Park et al., 1995) and the methodology for generating ARs from the *apriori* algorithm (Agrawal and Srikant, 1994) was proposed. The experimental results prove that this approach precisely reflects the information hidden in the datasets, and on top of it, as the dataset increases, it scales-up linearly in terms of processing time and memory usage.

On the other hand, the work realised by Aumann et al. in (Aumann and Lindell, 1999), proposes a new definition for QARs. An example of this rule would be: $sex = female \Rightarrow Wage : mean = \$7.90\ p/hr$ (overall mean wage = $\$9.02$). This form of QAR, unlike others doesn't require the discretisation of attributes with real number domains as a pre-processing step. Instead it uses the statistical theory and data-driven algorithms to process the data and find regularities that lead to the discovery of ARs. A step forward in this kind of rules was given by (Okoniewski et al., 2001). They provide variations of the algorithm proposed in (Aumann and Lindell, 1999) enhancing it by using heuristic strategies and advanced database indexing. The whole methodology is completed with the proposition of post-processing techniques with the use of similarity and significance measures.

The motivation of this work is to tackle some of the drawbacks of the previous techniques. Most of them require the translation of the original database so that each non-binary attribute can be regarded as a discrete set of binary variables over which the existing data

mining algorithms can be applied to. This approach can be sometimes unsatisfactory due to the following reasons: the translated database can be larger than the original one, the transformation of the quantitative data could not correspond to the intended semantics of the attributes. Moreover, current approaches do not deal with heterogeneous attributes but define ad-hoc solutions for particular data types (mainly numerical ones). As a consequence, they do not provide a common data mining framework where different representations, interesting measures and value clustering techniques can be properly combined.

1.1 Overview of Our Proposal

In this article, we extend the work introduced in (Danger et al., 2004) by applying clustering techniques in two steps of the mining process. A schematic view of the overall process can be seen in Figure 1. First, clustering is applied to the attribute domains, so that each object can be expressed as a set of pairs $\langle attribute, cluster \rangle$ instead of $\langle attribute, value \rangle$. This new representation allows users to define the most appropriate technique to discretize numeric domains or to abstract categorical values. We name *object subdescription* to the characterisation of an object through value clusters. The second step consists of clustering *object subdescriptions* in order to find frequent patterns between their features.

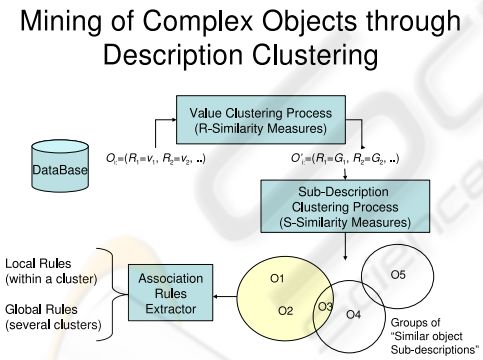

Figure 1: Overview of our proposal.

Finally, we propose an algorithm capable of obtaining the frequent itemsets from the found object subdescription clusters. We distinguish two kind of ARs, namely: inter and intra-cluster. The former relate attributes of different clusters, whereas the latter relate attributes locally defined in a cluster. Both kind of ARs provide different levels of details to users, which can mine a selected cluster involving a restricted set of attributes (i.e. local analysis) or the whole set of clusters (i.e. global analysis).

The paper is organised as follows: in the next section, we introduce the necessary concepts of the proposed framework. Then, in Section 3 we explain how we include clustering in our mining process. In Section 4 we describe a data-mining algorithm that finds frequent object sub-descriptions, and in Section 5 we describe the preliminary experimental results. Finally, in Section 6 we give our conclusions and we outline the future work.

2 FORMAL DEFINITIONS

In the proposed framework, a data collection consists of a set of objects, $\Omega = o_1, o_2, ..., o_n$, which are described by a set of features $R = R_1, R_2, ..., R_m$. We will denote with D_i the domain of the i-th feature, which can be of any data type.

We will apply the clustering algorithm to the attributes' domains in order to find groups (clusters) of close values and use them instead of the original values. Thus each object will be no longer characterised by it's attributes' values but by the clusters to which these values belong. We will denote the set of clusters in the domain (D_i) of a given attribute i as $\Pi_i = G_{i,1}, ..., G_{i,r}$, being $r \geq 1$ and $G_{i,r}$ the r-th cluster in the domain of the i-th attribute.

On the other hand, we will apply a second clustering step to the object sub-descriptions in order to generate groups of objects that will help us in reducing the final number of rules. We will denote the set of clusters in Ω as $\Theta = OG_1, ..., OG_t$, being $t \geq 1 \leq n$ and OG_i the i-th cluster in the objects' domain.

In order to compare two attribute-clusters, each feature R_i has associated a *comparison criterion*, $C_i(x, y)$, which indicates whether the pair of clusters, $x, y \in \Pi_i$, must be considered equal or not. This comparison criterion can include specifications for the case of invalid and missing values in order to deal with incomplete information.

The simplest comparison criterion is the strict equality, which can be applied to any domain:

$$C(x,y) = \begin{cases} 1 & \textit{if } x = y \\ 0 & \textit{otherwise} \end{cases}$$

Another interesting criteria can use the centroid of each domain cluster. For example, being $c_{i,r}$ the centroid of the r-th cluster over the i-th attribute the comparison function looks as follows:

If $x \in G_{a,1}$ and $y \in G_{a,2}$ then

$$C(x,y) = \begin{cases} 1 & \textit{if } |c_{a,1} - c_{a-2}| \leq \epsilon \\ 0 & \textit{otherwise} \end{cases}$$

Which expresses the fact that two clusters are considered equal if their centroids differ from each other in at most a given threshold ϵ.

Since the mining process is intended to discover

the combinations of object features and object clusters that frequently co-occur, it is necessary to manage the different object projections. Thus, a *subdescription* of an object *o* for a subset of features $S \subseteq R$, denoted as $I|_S(o)$, is the projection of o over the feature set S. In this context, we denote $o[r]$ the value of the object o for the feature r.

Moreover, we assume that there exists a *similarity function* between two object subdescriptions, which allow us to decide whether two objects o_i and o_j must be considered equal or not by the mining process. All the similarity functions are binary, that is, they return either 0 (not equal) or 1 (equal).

The simplest similarity function is the following one:

$$Sim(I|_S(o), I|_S(o')) = \begin{cases} 1 & \text{if } \forall r \in S, C(o[r], o'[r]) = 1) \\ 0 & \text{otherwise} \end{cases}$$

which expresses the strict equality by considering the comparison criterion of each of the subdescription features.

Alternatively, the following similarity function states that two subdescriptions are considered equal if they have at least ϵ features belonging to the same cluster:

$$Sim(I|_S(o), I|_S(o')) = \begin{cases} 1 & \text{if } |\{r \in S | C(o[r], o'[r]) = 1\}| \geq \epsilon \\ 0 & \text{otherwise} \end{cases}$$

In order to compare object-clusters, we can take one *representative* object of each cluster. In our approach, such a representative corresponds to the object with maximum connectivity according to the adopted similarity function. This is because we use a clustering algorithm that generates star-shaped clusters.

Analogously to the traditional Data Mining works, we also provide definitions of *support* and ARs, but applied to this new context.

We define the *support* of a *subdescription* $v = I|_S(o)$, denoted with $Sup(v)$, based in the work by (Danger et al., 2004), as the percentage of objects in Ω whose subdescriptions are similar to v, that is:

$$Sup(v) = \frac{|\{o' \in \Omega | Sim(I|_S(o'), v) = 1\}|}{|\Omega|}$$

We say that a pair of *subdescriptions* $v_1 = I|_{R_1}(o)$ and $v_2 = I|_{R_2}(o)$, with $R_1 \bigcap R_2 = \emptyset$ and $R_1, R_2 \subset R$, are associated through the AR $v_1 \Rightarrow v_2(s, c)$, if $Sup(v') \geq s$ and $\frac{Sup(v')}{Sup(v_1)} \geq c$, where $v' = I|_{R_1 \cup R_2}(o)$. The values of s and c are called *support* and *confidence* of the rule respectively.

The problem of computing the AR for complex objects consists of finding all the AR of the *subdescriptions* of Ω whose *support* and *confidence* satisfy the user-specified thresholds.

It must be pointed out that the previous definitions subsume the traditional concept of AR, therefore, if we use strict equality in both the comparison criterion and the similarity function, we obtain the classical definition of AR.

Besides, we can include other comparison criteria such as the interval-based partitions, for quantitative data, and the *is-a* relationship of the concept taxonomies, in order to represent other kinds of ARs (Srikant and Agrawal, 1997) (Z. Zhing and Zhang, 1997) (Hipp et al., 1998).

The idea that different items have different levels of interest for the user, as suggested in (Gyenesei, 2000), can be also incorporated in this framework by assigning a weight to each variable in the similarity function. Moreover, when the variables' data is fuzzy, it is perfectly admissible to use as a comparison criterion the membership of the values to the same fuzzy set.

3 FINDING INTERESTING SUBDESCRIPTIONS

In a previous step to that of finding the interesting ARs we will pre-process the data by means of clustering algorithms in order to find the groups that will be the base of our mining process.

The objective of this pre-processing step is that of identifying clusters in the domain of the attributes that will characterise the objects we will use to extract intra-cluster rules, and identifying clusters in the domain of the recently discovered object subdescriptions in order to extract inter-cluster rules.

The algorithm chosen for this process is the Star Clustering Algorithm introduced in (Aslam et al., 1998), and modified to be order independent in (Gil-García et al., 2003). The main reason for choosing it is that the Star-based representation of the objects subdescriptions seems a good way of representing the support of each subdescription (i.e. the number of objects that are similar to it, also called, satellites, as we will see later). Briefly, the star-shaped graphs capture the most supported subdescriptions w.r.t. the defined similarity function.

This algorithm approximates the minimal dominant set of the $\beta - similarity$ graph. The minimal dominant set (Kann, 1999) is the smallest set composed of graph's vertexes that contains every vertex in the graph, or at least if a vertex is not contained, it has a neighbour that does. The members of the minimal dominant set are called *stars* and their neighbours *satellites*.

A star-shaped sub-graph of $l + 1$ vertexes consists of a star an l satellites. Each sub-graph forms a group and the *stars* are the objects with the biggest connectivity. If an object is isolated in the graph it is considered as well a *star*.

The basic steps of this algorithm are the following ones:

- Obtain the $\beta - similarity$ graph.
- Calculate the degree of every vertex.
- While there's still ungrouped sub-vertexes do:
  - Take the ungrouped vertex with the highest degree.
  - Build a group with it an its neighbours.

Figure 2 shows the star-shaped graph of a cluster of object subdescriptions. The complexity of the algorithm is in $O(n^2)$, being n the number of processed objects.

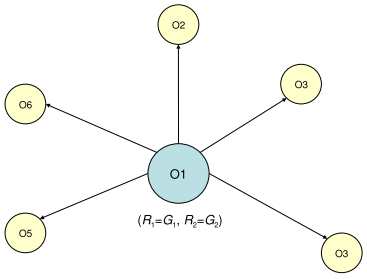

Figure 2: Example of star-based object cluster.

4 EXTRACTING ASSOCIATION RULES

In this section we present an algorithm (see Figure 3) for computing the frequent subdescriptions for a collection of complex objects. This algorithm is inspired in the original algorithm of (Agrawal and Srikant, 1994). However, it also uses the strategy of the Partition algorithm (Srikant and Agrawal, 1997) to compute the *support* of object subdescriptions.

It is worth mentioning that in this work an itemset is a *subdescription*, and its *support* is the number of objects in the database that are similar to it.

The algorithm works as follows: first, it determines all the groups of values for each feature by using the *SetFreqClusters* function, which applies the specific clustering criterion and similarity function defined for it. Then, while at least two groups have been found in the previous iteration *k*, they are combined two by two in order to create candidate sets of $k + 1$ features. Afterward, it determines, for each combination of variables, which of the candidate subdescriptions are frequent enough.

It's important to take into account that in order to guarantee the monotonic construction of the frequent itemsets, it is necessary that the similarity functions satisfy the folowing condition: if two objects are different with respect to a subdescription S_1, they are also different with respect to any other subdescription S_2, such that $S_1 \subset S_2$ (Danger et al., 2004).

Table 1: Features of the "Flags of the World" database.

Example database		
Feature Number	Feature name	Domain
1	Colours	Set of Colours[a]
2	Religion	Religions[b]
3	Number of Colours	Integer
4	Continent	Continents[c]
5	Number of vertical bars	Integer
6	Number of horizontal stripes	Integer
7	Number of sun or star symbols	Integer
8	Number of circles	Integer
9	Predominant colour	Set of Colours[a]
10	Colour in the top-left corner	Set of Colours[a]
11	Colour in the bottom-left corner	Set of Colours[a]
12	Geographic quadrant	NE, SE, SW, NW

[a] yellow, gold, red, green, blue, brown, orange, white, black

[b] Catholic, Other Christian, Muslim, Buddhist, Hindu, Ethnic, Marxist, Others

[c] North America, South America, Europe, Africa, Asia, Oceania

5 PRELIMINARY RESULTS

In this section we will give an example of the type of ARs that are extracted from a database once applied the mining process.

As earlier mentioned, we apply the mining algorithm in two ways, intra- and inter-cluster. In order to give examples of this kind of rules, we have taken the well-known "Flags of the world" database[1], which is summarized in Table 1.

We will use the same notation as in the formal framework for the different clusters, being for example, $G_{1,2}$, 2nd cluster in the domain of the 1st variable and OG_2 the 2nd cluster of object subdescriptions.

5.1 Intra-cluster Rules

Let us suppose that we define the following clusters for the different feature domains:

- Colour (colors in the flag): $G_{1,1} = (white, red)$, $G_{1,2} = (blue)$, $G_{1,3} = (green)$.

[1] http://www.cia.gov/cia/publications/factbook/docs

FreqItemSets_ComplexObjects(Ω, CriterionComps, SimilFuncs, MinSupp, FreqSets)

Input: $\Omega = \{o_1, o_2, ..., o_n\}$, a set of complex objects.
CriterionComps: array of comparison's functions. The i-th component in the array corresponds to the comparison criterion for the i-th feature.
SimilFuncs: Dictionary of similarity functions, such that the key that corresponds to the similarity function for the subdescription $S' = \{K_{i_1},...,K_{i_s}\}$ is the own set *S'*.
MinSupp: Minimal support to consider a subdescription as frequent.

Output: FreqSets: Set of dictionaries that maintains for each size and combination of features (with at least one frequent cluster) the frequent sub-descriptions in Ω and the index of the objects that are similar to each one of these sub-descriptions.

Method:

$F_1 = SetFreqClusters(\Omega, CriterionComps)$
$k = 2$
while $F_{k-1} \neq \emptyset$ do
  $SetCandidatesVars = \{\{f_i, f_j\} / f_i, f_j \in F_{k-1}.keys(), |f_i \cup f_j| = k\}$
  for each pair of features $\{f_i, f_j\}$ in $SetCandidatesVars$ do
    for each $O \in \Omega$ do
      if $I|_{f_i}(O) \in F_{k-1}[f_i].keys()$ and $I|_{f_j}(O) \in F_{k-1}[f_j].keys()$ then
        $IndexSimObjs = F_{k-1}[f_i][I|_{f_i}(O)] \cap F_{k-1}[f_j][I|_{f_j}(O)]$
        $SimObjs = \{\}$
        for $O_k \in \Omega, k \in IndexSimObjs$ do
          if $SimilFuncs[f_i \cup f_j](I|_{f_i \cup f_j}(O), I|_{f_i \cup f_j}(O_k)) = 1$ then
            $SimObjs = SimObjs \cup \{k\}$
        if $|SimObjs| \geq MinSupp$ then
          $F_k[f_i \cup f_j][I|_{f_i \cup f_j}(O)] = SimObjs$
  $FreqSets = FreqSets \cup \{F_k\}$
  $k = k + 1$

Figure 3: Data Mining Algorithm.

Table 2: Object clusters found by the Star-based clustering algorithm.

Object Clusters		
Object Cluster	SUBDESCRIPTIONS	#OBJ
OG_1	$(G_{1,1}, G_{2,1}), (G_{1,1}, G_{2,2})$	20
OG_2	$(G_{1,3}, G_{2,3}), (G_{1,2}, G_{2,3})$	10
OG_3	$(G_{3,2}, G_{1,1}), (G_{3,2}, G_{1,2})$	15
OG_4	$(G_{3,1}, G_{1,3}), (G_{3,1}, G_{1,2})$	30
OG_5	$(G_{4,4}, G_{1,3}), (G_{3,1}, G_{1,3}),$ $(G_{3,2}, G_{1,3})$	20
OG_6	$(G_{3,2}, G_{1,1}, G_{6,1}, G_{5,1}),$ $(G_{3,2}, G_{1,2}, G_{6,1}, G_{5,1})$	40

- Religion (majority religion in the country): $G_{2,1} = (Catholic)$, $G_{2,2} = (OtherChristian)$, $G_{2,3} = (Other)$.
- Number of Colours (number of different colors present in the flag): $G_{3,1} = (2)$, $G_{3,2} = (3, 4)$.
- Continent (Continent to which the country belongs): $G_{4,1}$ ="North America", $G_{4,2}$ ="South America", $G_{4,3}$ ="Europe", $G_{4,4}$ ="Africa", $G_{4,5}$ ="Asia", $G_{4,6}$ ="Oceania".
- Number of vertical bars: $G_{5,1} = (1)$, $G_{5,2} = (2)$, $G_{5,3} = (3)$, $G_{5,4} = \{i | i > 3\}$.
- Number of horizontal stripes: $G_{6,1} = (1)$, $G_{6,2} = (2)$, $G_{6,3} = (3)$, $G_{6,4} = \{i | i > 3\}$.
- Number of star and sun symbols: $G_{7,1} = (1)$,

$G_{7,2} = (2)$, $G_{7,3} = (4)$, $G_{7,4} = (5)$, $G_{7,5} = \{i|i > 5\}$.

- Number of circles: $G_{7,1} = 1$, $G_{7,2} = 2$, $G_{7,3} = 3$, $G_{7,4} = 4$, $G_{7,5} = \{i|i > 4\}$.
- Predominant color: $G_{9,1}$ ="yellow", $G_{9,1}$ ="gold", $G_{9,1}$ ="red", $G_{9,1}$ ="green", $G_{9,1}$ ="blue", $G_{9,1}$ ="brown", $G_{9,1}$ ="orange", $G_{9,1}$ ="white", $G_{9,1}$ ="black".
- Colour in the top-left corner: $G_{10,1}$ ="yellow", $G_{10,1}$ ="gold", $G_{10,1}$ ="red", $G_{10,1}$ ="green", $G_{10,1}$ ="blue", $G_{10,1}$ ="brown", $G_{10,1}$ ="orange", $G_{10,1}$ ="white", $G_{10,1}$ ="black".
- Colour in the bottom-left corner: $G_{11,1}$ ="yellow", $G_{11,1}$ ="gold", $G_{11,1}$ ="red", $G_{11,1}$ ="green", $G_{11,1}$ ="blue", $G_{11,1}$ ="brown", $G_{11,1}$ ="orange", $G_{11,1}$ ="white", $G_{11,1}$ ="black".
- Geographic quadrant: $G_{12,1}$ ="NE", $G_{12,2}$ ="SE", $G_{12,3}$ ="SW", $G_{12,4}$ ="NW".

We obtain the following rules from two of the detected subdescription clusters (see Table 2):

- From OG_3: *Number of Colours* $= G_{3,2}$, $Colour = G_{1,1}(31\%, 80\%)$. Meaning that in the 80% of the cases, if a flag contains 3 or 4 different colors, one of them is either red or white. This rule has a support of 31%.
- From OG_1: $Religion = G_{2,2}, Colour = G_{1,1}(21\%, 68\%)$. Meaning that in the 68% of the cases, if a country's majoritary religion is the Christian (other than the Roman Catholic), its flag contains red or white. This rule has a support of 21%.
- From OG_3: $Colour = G_{1,2}, Colour = G_{1,1}(32\%, 63\%)$. The color blue implies the presence of both colors red and white in the 63% of cases.
- From OG_6: *Number of Colours* $= G_{3,2}$, $HorizontalStripes = G_{6,1}, VerticalBars = G_{5,1}(20\%44\%)$. In 44% of the cases a flag containing 3 or 4 colors is composed by one quadratic section.

5.2 Inter-cluster Rules

Inter-cluster rules involve object clusters that satisfy the following conditions:

- the intersection between their member sets is greater than the minimum support value $MinSupp$ and
- they do not share some of their features.

For example, from Table 2, the following pairs are candidate to be mined for finding inter-cluster rules: (OG_1, OG_3), (OG_2, OG_4),(OG_2, OG_5),(OG_3, OG_4),(OG_3, OG_6) and (OG_4, OG_5). For each of these pairs, the mining algorithm calculates all the frequent object subdescriptions. For example, the following rules have been extracted from the previous cluster pairs:

- From (OG_1, OG_3): In 30% of the cases, the countries where the majoritary religion is the Christian, their flags have the colors White, Red or Blue.
- From (OG_2, OG_4): In 40% of the cases, countries in which the majoritary religion is other than the Christian,their flags have less than 3 colors, being one of them either blue or green.
- From (OG_2, OG_5): In 30% of the cases, the African countries have a religion different from the Christian.
- From (OG_3, OG_6), In 35% of the cases, the flags with just one quadratic section contain Blue, Red and White.

Notice that the same mining algorithm of Figure 3 is applied to find both inter- and intra-cluster association rules. The difference consists of the set of subdescription objects that is used as input. For local analysis, just the members of a single object cluster is passed to the algorithm. Instead, for global analysis, the union of the members of a set of related candidate clusters is passed to the algorithm.

6 CONCLUSIONS AND FUTURE WORK

This paper presents a general framework for mining complex objects represented with any of the existing data models (e.g. relational, object-oriented and semi-structured data models). The mining process is guided by the semantics associated to each object description feature (attributes), which are stated by the users by selecting the appropriate representation model. This was the model introduced by (Danger et al., 2004). Furthermore, we have extended the framework to enrich the formal representation of the objects using clusters of both attributes and objects, so that the mining process results in an acceptable number of higher level rules. We show as well examples of this semantically richer rules. The future work includes carrying out a series of experiments over well-known databases (e.g ftp://ftp.ics.uci.edu/pub/machine-learning-databases/) and the Monalisa repository database (http://alimonitor.cern.ch:8889), which is the Grid monitoring database for the ALICE experiment at CERN, in order to prove that the proposed method is generating the expected results.

REFERENCES

Agrawal, R., Imielinski, T., and Swami, A. N. (1993). Mining association rules between sets of items in large databases. In Buneman, P. and Jajodia, S., editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C.

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. In Bocca, J. B., Jarke, M., and Zaniolo, C., editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann.

Aslam, J. A., Pelekhov, K., and Rus, D. (1998). Static and dynamic information organization with star clusters. In *CIKM*, pages 208–217.

Aumann, Y. and Lindell, Y. (1999). A statistical theory for quantitative association rules. In *KDD*, pages 261–270.

Danger, R., Ruiz-Shulcloper, J., and Berlanga, R. (2004). Objectminer: A new approach for mining complex objects. In *ICEIS (2)*, pages 42–47.

Dong, L. and Tjortjis, C. (2003). Experiences of using a quantitative approach for mining association rules. In *IDEAL*, pages 693–700.

Gil-García, R., Badía-Contelles, J. M., and Pons-Porrata, A. (2003). Extended star clustering algorithm. In *CIARP*, pages 480–487.

Gyenesei, A. (2000). Mining weighted association rules for fuzzy quantitative items. In *PKDD '00: Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 416–423, London, UK. Springer-Verlag.

Hipp, J., Myka, A., Wirth, R., and Güntzer, U. (1998). A new algorithm for faster mining of generalized association rules. In *Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD '98)*, pages 74–82, Nantes, France.

Kann, V. (1999). *A compendium of NP optimization problems. In Complexity and Aproximation*. Springer Verlag.

Kuok, C. M., Fu, A. W.-C., and Wong, M. H. (1998). Mining fuzzy association rules in databases. *SIGMOD Record*, 27(1):41–46.

Miller, R. J. and Yang, Y. (1997). Association rules over interval data. pages 452–461.

Okoniewski, M., Gancarz, L., and Gawrysiak, P. (2001). Mining multi-dimensional quantitative associations. In *INAP*, pages 265–274.

Park, J. S., Chen, M.-S., and Yu, P. S. (1995). An effective hash based algorithm for mining association rules. In Carey, M. J. and Schneider, D. A., editors, *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pages 175–186, San Jose, California.

Srikant, R. and Agrawal, R. (1996). Mining quantitative association rules in large relational tables. In Jagadish, H. V. and Mumick, I. S., editors, *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 1–12, Montreal, Quebec, Canada.

Srikant, R. and Agrawal, R. (1997). Mining generalized association rules. volume 13, pages 161–180.

Tong, Q., Yan, B., and Zhou, Y. (2005). Mining quantitative association rules on overlapped intervals. In *ADMA*, pages 43–50.

Z. Zhing, Y. L. and Zhang, B. (1997). An effective partitioning-combining algorithm for discovering quantitative association rules. In *Proc. of the First Pacific-Asia Conference on Knowledge Discovery and Data Mining*.