Discovering diverse association rules from multidimensional schema

Abstract

The integration of data mining techniques with data warehousing is gaining popularity due to the fact that both disciplines complement each other in extracting knowledge from large datasets. However, the majority of approaches focus on applying data mining as a front end technology to mine data warehouses. Surprisingly, little progress has been made in incorporating mining techniques in the design of data warehouses. While methods such as data clustering applied on multidimensional data have been shown to enhance the knowledge discovery process, a number of fundamental issues remain unresolved with respect to the design of multidimensional schema. These relate to automated support for the selection of informative dimension and fact variables in high dimensional and data intensive environments, an activity which may challenge the capabilities of human designers on account of the sheer scale of data volume and variables involved. In this research, we propose a methodology that selects a subset of informative dimension and fact variables from an initial set of candidates. Our experimental results conducted on three real world datasets taken from the UCI machine learning repository show that the knowledge discovered from the schema that we generated was more diverse and informative than the standard approach of mining the original data without the use of our multidimensional structure imposed on it.

Keywords: Association rules; Data cubes; Knowledge discovery; OLAP analysis; Multidimensional schema

1. Introduction

Data mining enables the discovery of hidden trends from large datasets, while data warehousing provides for interactive and exploratory analysis of data through the use of various data aggregation methods. Both technologies have essentially the same set of objectives and can potentially benefit from each other's methods to facilitate knowledge discovery. Each technology is mature in its own right, and despite the very clear synergy between these two technologies, they have developed largely independent of each other. More recently, there has been an increased research interest in the data mining community towards integrating the two technologies (Goil and Choudhary, 2001, Usman and Asghar, 2011, Usman and Pears, 2010, Liu and Guo, 2001, Ohmori et al., 2007, You et al., 2001, Zhen and Minyi, 2001, Usman et al., 2009). However, most of this body of work concentrated on applying data mining as a front end technology to a data warehouse in order to extract trends and patterns from data resident in data warehouses.

There has been relatively less research in leveraging data mining techniques in the design of data warehouses or multidimensional schema (Zubcoff et al., 2007, Sapia et al., 1999, Pardillo and Mazón, 2010, Pardillo et al., 2008, Usman et al., 2010). Yet, there remains a need for automated support in the design of data warehouses, especially in domains containing high dimensional data. In such domains the sheer scale of the data, both in terms of data volume as well as in the number of dimensions, may make it difficult for human designers to decide which dimensions are the most informative and should thus be retained in the final version of the cube design.

(Usman and Pears, 2011b) used a hierarchical clustering technique in conjunction with multidimensional scaling (Cox and Cox, 2008) to design schema at different levels of data abstraction. They developed an iterative method that explores the similarities and differences in information contained across consecutive levels in the cluster hierarchy. The presentation of such information at different levels of abstraction provides decision makers with a better understanding of the patterns and trends present in the data. While their work does provide a basis for exploiting data mining methods in the design of multidimensional schema, a number of issues remained unresolved.

Firstly, in high dimensional data environments, the sheer size and number of data cubes to be explored makes data analysis a laborious and potentially error prone activity, which in turn adversely affects the quality of the design process.

Secondly, in high dimensional environments the design and data analysis processes need to be integrated with each other. With the use of appropriate information theoretic measures such as Entropy in the design process, less informative dimensions can be filtered out, thus leading to a more compact, useful and manageable schema. In terms of data analysis, the main tool used in multidimensional analysis in a data warehousing environment is the use of various data aggregation and exploratory techniques that form part of the On Line Analytical processing (OLAP) suite of methods. While traditional OLAP methods are excellent tools for exploratory data analysis they are limited as far as detecting hidden associations between items resident in a data warehouse. The discovery of such hidden relationships and associations often yields important insights into underlying trends and in general leads to an improved decision making capability.

OLAP is primarily used for exploratory data analysis and its capabilities are limited not only in detecting hidden associations between items but also predicting and forecasting interesting items in data cubes. Predictive analytics is being demanded for enterprise management and, is often required by decision makers when strategic decisions need to be taken. Recently, new approaches are attempting to extend OLAP to prediction capabilities, in order to anticipate and forecast future events (Agarwal and Chen, 2011, Abdelbaki et al., 2012). In this paper, we present an approach to extend the capability of OLAP to detect hidden associations and to predict future trends based on historical data by introducing association rule mining on multidimensional schema.

Association rule mining (Agrawal et al., 1993) aims to extract interesting correlations and associations among sets of items in transaction databases. Let $I = \{i_1, i_2, ..., i_n\}$ be a set of items. Given a set of transactions *D*, where each transaction T is a set of items such that $T \subseteq I$, an association rule is of the form $A \Rightarrow B$ where $A \subseteq I,$ and $B \subseteq I,$ and $A \cap B = \phi.$ A and B are the called antecedent (left hand side) and consequent (right hand side) of the rule. An example of a common association rule is bread → butter. This indicates that a customer buying bread would also buy butter under user defined threshold of confidence and support. The confidence of the rule measures the degree of correlation between itemsets, while support measures the significance of the correlation between itemsets. Thus, association rule mining is a process of finding all rules that are greater than the user defined minimum support and minimum confidence.

In this research we have made the following main contributions:

- Proposed a knowledge discovery methodology that utilizes a combination of machine learning and statistical methods to identify interesting regions of information and diverse association rules in large multidimensional data cubes.
- Provided an algorithm for constructing a binary tree from hierarchical clustering results (dendrogram).
- Proposed a measure based on Information Gain to identify and rank the most informative dimensions among nominal variables that should be retained for schema design.
- Applied well-known dimension reduction techniques such as Principal Component Analysis (PCA) in order to identify and rank the most informative numeric facts present in high dimensional datasets.
- Generated informative cubes at different levels of data abstraction and studied the effect of abstraction level on information content.
- Provided methods to construct candidate schema with highly ranked dimensions (nominal variables) and measures (numeric variables).
- Performed case studies on three real-world datasets to validate our methodology and showed that it enables analysts to find cubes of interest and the diverse association rules. Furthermore, we showed that rules generated from our semi-automatically generated multidimensional schema are in general more diverse and have better predictive accuracy than rules generated from the same data without the use of the multidimensional schema.

The rest of the paper is organized as follows. Section 2 presents an overview of previous research relevant to the design and analysis of multidimensional schema. In Section 3 we review the methodology that underpins our approach to design and analysis of multidimensional data. The extensions proposed to this basic methodology are covered in Section 4, along with a running example that illustrates each step. Section 5 presents the application of the proposed methodology on three case studies. Finally, a summary of research achievements and some possible research directions for future research are discussed in Section 6.

2. Related work

In this section, we review two main themes of literature related to the contributions made in this paper. We present work in automating the design and construction of multidimensional warehouse schema followed by the application of mining techniques to enhance knowledge discovery from multidimensional data.

2.1. Automating the design and construction of multidimensional schema

A number of schema design approaches have been proposed in the literature to provide automated support for the design of multidimensional schema. Pardillo et al. (2010) identified that most of the research in schema design focuses on the automatic derivation of database schemata from conceptual models but does not address the problem of design of multidimensional schema. Pardillo et al. emphasized that the main issue in data warehouse construction is multidimensional modelling. To resolve this issue, they proposed a Model Driven Architecture Framework (MDA) approach for multidimensional modelling. In their approach, they built different MDA models using an extension of the Unified Modelling Language (UML) and Common Warehouse Meta-model (CWM) (Poole and Mellor, 2001) to formally establish transformations between MDA models using a query language. UML integration with CWM allows end-user tools to query the multidimensional schema accurately and reduce design/development time.

Likewise, Dori et al. (2008) suggested an Object-process-based Data Warehouse Construction Method (ODWC) for designing multidimensional schema. Dori et al. suggested that the suitability of current multidimensional modelling methods for large-scale systems is questionable, as they require multiple manual actions to discover measures and relevant dimensional entities and they tend to disregard the system's dynamic aspects. They proposed the ODWC method which utilizes the conceptual model of operational systems to construct a corresponding multidimensional schema. The method operates by first selecting business processes and models them in the form of snowflake schemas. Secondly, it selects the schema that is most appropriate for the organization’s data mining needs. The main limitation of the ODWC method is the strong assumption that business processes are well defined by the organization. There could be cases in which the organization’s business processes are not well established or not aligned with the organization’s data mining needs. In such cases, the proposed method will require additional manual actions to configure business processes for the generation of meaningful snowflake schemas.

A similar semi-automatic technique has been proposed by Palopoli et al. (2002) for generating multidimensional schema from operational databases. The proposed technique takes in a list of database schemas in the form of an Entity Relationship model and a dictionary of lexical synonymy properties to generate a schema as an output. However, this technique has been tested on a single case study and requires further cases to be studied before it can be considered as a comprehensive modelling technique with wider applicability. To accomplish ease in schema design, Tryfona et al. (1999) built a conceptual model (*StarER*) for multidimensional modelling on the basis of user modelling requirements. The *StarER* model combines the star structure, which is dominant in data warehouses, with the semantically rich constructs of the ER model. Examples from a mortgage data warehouse environment, in which *StarER* has been tested, revealed the ease of understanding the model, as well as its efficiency in representing complex information at the semantic level.

In the quest to propose a generic modelling technique, Hahn et al. (2000) proposed the generation of tool specific OLAP schemata from conceptual graphical models. A new approach named *Bablefish*, has been suggested to generate multidimensional schema. It allows graphical representation of a conceptual schema for interactive modelling purposes. Furthermore, the proposed approach discusses the issue of translating graphical representations to configurations for real-world OLAP tools and introduces a *View* concept that allows designers to model interconnections of static schema with other aspects of warehouse design such as transformation modelling, data source modelling, security modelling etc. The main benefit of the approach is its applicability to Greenfield situations when no system is already in place. In real world development, there exist cases in non-business domains (Mansmann, 2009) where the data warehouse developer needs to design schema where an operational system does not exist. Furthermore, such domains may be ill-defined domains (Nkambou et al., 2011), thus compounding the difficulty of the design problem. Therefore, their research is similar to the work undertaken in this paper as we also target those cases where limited domain knowledge exists and no operational system is in place. However, the scope of our work is wider as we not only generate schema but also equip analysts to discover knowledge from the generated schema.

Peralta et al. (2003) stated that design automation usually focuses on data models, data structures and criteria for defining table partitions and indexes. A rule-based mechanism was proposed to automate the design of multidimensional schema. A set of design rules, embedding design strategies, decide the application of suitable transformations in order to generate logical multidimensional schema. The proposed system has been prototyped, which applies design rules by using an algorithm that considers frequent design problems suggested in existing methodologies. Likewise, an automatic tool for generating a star schema from an Entity-Relationship Diagram (ERD) was introduced by Song et al. (2008). A prototype named *SAMSTAR* was presented, which was used for the automatic generation of star schema from an ERD. With this automatic generation of star schema, the system helps designers reduce their effort and time in building data warehouse schemas.

More recently (Usman et al., 2013) proposed a methodology for the design of multidimensional schema and discovery of interesting cube regions in multidimensional datasets. The distinctive feature of their approach is the use of robust data reduction methods such as PCA and Multiple Correspondence Analysis (MCA) to identify variables that capture the greatest degree of variation in high dimensional data. Such variables were utilized to design schema and construct informative data cubes which contain interesting regions of information. These informative data cubes were generated at different levels of data abstraction and the effect of abstraction level on information content was studied through OLAP analysis. However, OLAP analysis is limited to exploratory analysis and was not designed to discover interesting associations among data variables (Ben Messaoud et al., 2007). This limitation of OLAP motivated us in our current research to augment OLAP analysis with association rule mining methods to discover interesting relationships and associations among data variables at multiple levels of data abstraction.

It is apparent from the review in this section that the majority of previously proposed techniques, except (Usman et al., 2013), rely either on domain specific knowledge or existing operational systems to design multidimensional schema. To the best of our knowledge, none of the work done in the past focused on cases where limited or no domain knowledge exists. Additionally, there is a strong requirement to assist data warehouse designers to construct informative schema that can overcome design pitfalls and provide analysts a base to counterpart knowledge discovery challenges from large multidimensional space.

2.2. Enhanced knowledge discovery from multidimensional data

As cited by a number of authors (Ben Messaoud et al., 2007, Kaya and Alhajj, 2003, Nestorov and Jukic, 2003b), Kamber et al. (1997) were the first to target the issue of discovering associations in the form of rules in a multidimensional environment. In their proposed approach, a user specifies hypotheses in the form of meta-rules or pattern templates. A mining system then attempts to confirm the provided hypotheses by searching for patterns that match given meta-rules. The use of pattern templates ensures that rules found are of interest to the user. This method also has the advantage of making the rule discovery process efficient as the search for rules is conducted in a space constrained by the templates specified. However, the main drawback is that interesting rules that fall outside the template scope will not be discovered and this will happen when the user is unaware of unexpected and interesting patterns due to limited knowledge of the underlying data.

Zhu and Han in Zhu (1998) proposed an approach towards mining three types of multidimensional association rules, namely intra-dimensional, inter-dimensional and hybrid association rules. The proposed method leveraged OLAP technology to perform multi-level association rule mining on different levels of the dimensional hierarchy. However, interestingness evaluation of the generated rules was confined to correlation analysis of the left-hand side with the right-hand side of the rules and it is not clear how the rules would perform on other objective rule interest measures such as, for example, the rule diversity measure (Zbidi et al., 2006, Geng and Hamilton, 2006).

In order to generalize the methods of multidimensional association rule mining, Psaila et al. (2000) proposed a new approach to exploit the concept hierarchies present in the multidimensional model. With this approach, data miners reduce complexity in multidimensional data by exploiting concept hierarchies to guide the mining process towards potentially interesting mining queries. To estimate the degree of rule interestingness, the authors employed a metric that was specifically designed for analyzing sales data. The utility of rules generated against generic interest measures was not explored in this research.

Ng et al. (2002) focused on applying association rule mining to the most commonly used warehouse schema, the STAR schema. They proposed an efficient algorithm which makes use of the properties of the STAR schema and experimental results showed that the proposed algorithm significantly outperforms the conventional approach of joining the dimension tables first and then mining rules from the logically joined tables. This proposed joining process of merging the dimension tables present in the schema before mining rules, however, requires the data to be loaded from the data warehouse before applying association rule mining, which is computationally expensive when the volume of data stored is large.

In order to overcome this limitation, Chung and Mangamuri (2005) introduced another improved algorithm named Star-miner that can be implemented directly on the relational database system without the need for relational joins. However, their experimentation was only performed on synthetic data and hence there was no indication of how well the approach would work on real world data. A similar approach for mining the STAR schema was proposed in Nestorov and Jukic (2003b). The authors proposed a framework that enabled ad-hoc data mining queries to be run directly on the multidimensional warehouse structure. The proposed framework expanded the domain of the application of association rule mining from the transactional level to the aggregate data level. The use of dimensional information resulted in more detailed and actionable rules. Experimentation also showed that the rules could be generated much faster than with the conventional approach of generating rules from the transactional level. This research revealed new insights into the usefulness of extracting knowledge from multidimensional data vis-a-vis transactional data.

In order to mine association rules from data warehouses, Tjioe and Taniar (2005) proposed a pruning approach to filter non informative data in a data warehouse with the objective of discovering interesting rules. The authors proposed four algorithms *VAvg, HAvg, WMAvg* and *ModusFilter* which focus on pruning all rows present in the fact table that have less than the average quantity in data and provide *initialized tables*. After such pre-processing, the algorithms efficiently use these tables to mine association rules by focusing on summarized data present in the data warehouse. The results of performance evaluation show the effectiveness of the row pruning methods. Again, the rules generated through these proposed methods lack evaluation on other objective measures of interest such as the diversity measure.

Messaoud et. al in (Messaoud et al., 2006) highlighted another limitation in Tjioe's work, referring to the fact that it is limited only to the COUNT measure for mining rules from aggregated data. They proposed another method of mining rules based on measures such as *sum, avg, min* and *max*. They used criteria for the evaluation of rule importance such as *Lift* and *Loevinger* measures. This work has been extended by proposing an online environment for mining association rules called OLEMAR (Ben Messaoud et al., 2007). In the extended work visual representation has been added in order to visualize the importance of the discovered rules using Graphic Semiology principles. A real world case study conducted on a breast cancer dataset illustrated the efficiency and effectiveness of the proposed work. However, there is still a need to evaluate the performance of the discovered rules on objective measures of interest.

As with research in the area of multidimensional schema design, gaps exist in previous research in the area of knowledge discovery from multidimensional data. It is apparent from the review that all the previous approaches support constraining search space for rule discovery, either in the form of pattern templates or in the form of aggregated data. While aggregated data has been shown to be a better foundation for generating rules, the evaluation for the most part, with the exception of Messaoud et al. (2006), used support and confidence criteria to measure the importance of rules. These measures are best suited to transactional data but do not adequately measure the effectiveness of rules generated from data represented at different levels of data granularity. The complexity of multidimensional structures requires the use of more sophisticated measures to quantify the interestingness of rules discovered at different levels of data abstraction.

Finally, we observe that most of the work done in the past targeted the business domain and hence it would be of interest to investigate the effectiveness of rule discovery across non-business domains. Motivated by the above mentioned issues, we propose a generic methodology in this paper that provides automated assistance to constrain a multidimensional schema, supports advanced evaluation of discovered rules to measure interestingness, and offers easy implementation methods for non-business domains.

3. A multi-level approach to data warehouse design

In this section, we summarize the prior work of Usman and Pears (2011) which forms the foundation for our current research. The authors suggested that the expertise of a human data warehouse designer, with his/her limited knowledge of the domain may not be effective in high data volume and high dimensional environments. Furthermore, they pointed out that nominal variables, while being candidates for dimension variables, may not always be suitable candidates for use. This was for two reasons: firstly nominal variables which have low information content do not add value to the knowledge discovery process and could thus be excluded. Secondly, even when nominal variables have high information content it may not be appropriate to use them in raw form to define dimensions. This is typically the case in high cardinality nominal variables where the grouping of nominal values will lead to the discovery of more meaningful and useful patterns. They reasoned that the use of data mining techniques to aid in the discovery of meaningful dimensions could augment domain knowledge and thus enrich the design process.

They also pointed out that relationships between nominal and numeric variables may be subject to change, depending on the level of data granularity. To test this premise they applied a hierarchical clustering algorithm to the numeric variables to generate a dendrogram with nodes representing individual clusters containing a mix of numeric and nominal variables that are candidates for multidimensional schema. A multidimensional scaling method (Cox and Cox, 2008) was then applied on the nominal values within a cluster in order to transform them into numerical form. The motivation was to obtain a grouping of the nominal variables based on their pattern of co-occurrence within a given cluster. However, no concrete algorithm was proposed for deriving groups and it was left to the human designer's subjective judgement to decide boundaries between groups.

Despite the afore-mentioned contributions, their approach suffers from a number of limitations. Firstly, the methodology does not provide a clear indication of how many levels in the cluster hierarchy are required to optimize the knowledge discovery process. Human judgement is required to decide the cluster cut-off point and if this cut-off point is underestimated then valuable knowledge may be lost. On the other hand, overestimation leads to the situation of an unnecessarily large dendrogram with attendant space and computational inefficiencies. We address this problem in this paper by utilizing the linkage inconsistency threshold proposed by Cordes et al. (2002) to determine the cut-off point in a dendrogram. The use of a rigorous method of cut-off determination via the inconsistency threshold removes the need for the manual error-prone method of determination.

Secondly, a manual method was used for the extraction of clustered data and the labelling of clusters at the various levels of the hierarchy to generate a binary tree. This represents a laborious task for the analyst to extract data from each cluster and label each cluster, one by one. Besides the manual work of naming and extracting cluster data, users have to manually construct a binary tree structure in order to visualize the cluster hierarchy in the form of a hierarchical tree. We believe that cluster extraction, labelling and binary tree generation should be automated in order to ensure that knowledge discovery is efficient and robust. In this paper, we propose an algorithm that generates a binary tree of clusters based on an automatically identified cut-off point and labels clusters with automatically determined labels that are based on their position in the data hierarchy.

Thirdly, as mentioned previously, full automation for grouping of nominal variable is not provided. Instead, users are required to visualize the results of the multidimensional scaling technique in the form of a parallel coordinate display and to group similar values present in each dimension. Such grouping by visual inspection may not be feasible in cases where similar values lie very close to each other in a dimensional coordinate. Nominal variables with high cardinality such as *Country, Product codes* etc., having more than 40 distinct names, are difficult to visualize and group, and thus there is a need for a generic method that can create groups of similar values within each dimension automatically. In this work, we provide an algorithm which creates groups of similar values based on an automatically calculated threshold for each dimension.

Fourthly, no ranking mechanism for filtering non informative dimensions was provided. Users are required to decide the dimensions of their choice with no indication of the underlying information content. Thus it would be useful to provide guidance to users by ranking dimensions based on an objective information theoretic measure such as entropy and information gain, thus enabling users to factor in information content in addition to their specialized domain knowledge in the decision making process.

Finally, no explicit support was provided for the discovery of hidden relationships and associations which often yield important insights into underlying trends. To overcome the limitation of OLAP's incapability to find hidden associations, we applied association rule mining on our generated multidimensional schema. This enables the mining of hidden trends and patterns in the form of association rules from logically constrained schema. Moreover, we evaluate the interestingness of rules with respect to multiple objective rule interest measures proposed in Geng and Hamilton (2006) under the *diversity* criterion. We believe that rules containing diverse information convey more knowledge and hence, such rules are of more interest to the user.

4. Proposed methodology for multidimensional cube design and discovery of diverse rules

In this section, we present an overview of our methodology for multidimensional cube design that facilitates the discovery of diverse association rules. As mentioned in Section 3, the main objective is to equip knowledge workers with necessary information in order to assist in the cube design process and not to replace the specialized knowledge that domain specialists bring to bear in the designing of multidimensional schema. Figure 1 depicts the main steps of our proposed methodology.

We employ data mining and statistical techniques in conjunction with PCA to rank significant dimensions and facts. These highly ranked dimensions and facts lead to the discovery of interesting information embedded in multidimensional cubes. In step 1 a hierarchical set of clusters is obtained so that the data can be examined at different levels of data abstraction. Each of these clusters contains a mix of both numeric and nominal variables and steps 2 and 3 are used to rank the numeric and nominal variables respectively. In step 4, we apply multidimensional scaling technique in order to group the semantically related nominal values present in each nominal variable. These ranked lists and groupings are then utilized to generate a multidimensional schema for each cluster in step 5. The generated schema is then utilized in step 6 to construct informative data cubes for each cluster. Association rule mining is applied on the generated schema to detect association rules in step 7. Finally, the interestingness of the generated rules is evaluated on the basis of diversity criterion in step 8.

In the following sub-sections, we illustrate each step of our proposed methodology with the help of a hypothetical example. Consider a real-world mixed variable dataset ***D*** having 3 numeric (*Profit, Quantity, Weight*) and 3 nominal variables (*Quality, Color, Size*) with ***X*** number of records in it.

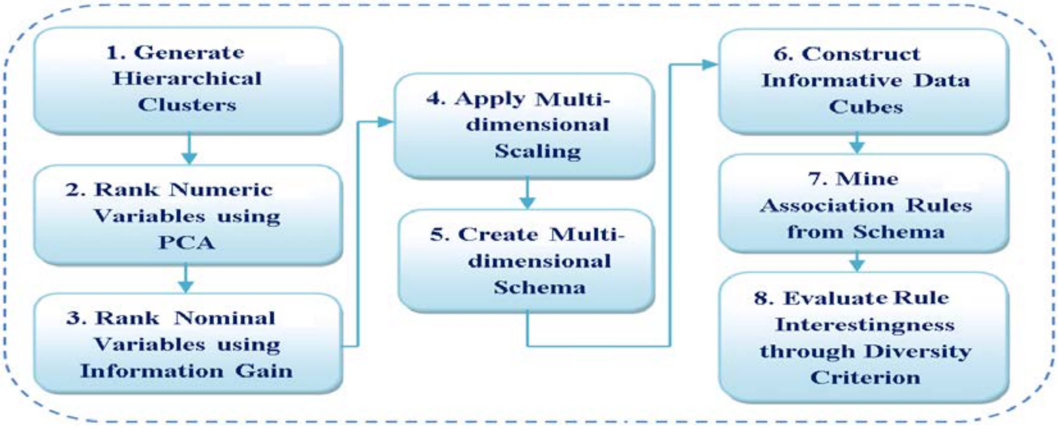

1. Generate Hierarchical Clusters
2. Rank Numeric Variables using PCA
3. Rank Nominal Variables using Information Gain
4. Apply Multi-dimensional Scaling
5. Create Multi-dimensional Schema
6. Construct Informative Data Cubes
7. Mine Association Rules from Schema
8. Evaluate Rule Interestingness through Diversity Criterion

Fig.1. Methodology for discovering diverse association rules

4.1. Generate hierarchical clusters

In the first step, we apply agglomerative hierarchical clustering on numeric variables of the given dataset to generate a dendrogram. Each level in the dendrogram contains a set of child clusters that were split off a single parent. One issue with any form of clustering is determining the number of clusters and with respect to hierarchical clustering this is reduced to determining at what point to terminate the generation of the dendrogram. We use the linkage inconsistency threshold (Cordes et al., 2002) to determine the cut-off point. The threshold is defined by equation 1.

$$ITh\left(link1\right)=\frac{length\left(link1\right)-\mu\left(all\,links\right)}{\sigma\left(all\,links\right)} \tag{1}$$

Figure 2 indicates the links and heights used in the calculation of the threshold defined in equation 1. The inconsistent coefficient uses the height information to calculate the mean of all links.

In equation 1, the distance between two clusters is represented as the length of the link, *link*1. The term μ represents the mean of all the links present in the dendrogram and σ is the calculated standard deviation across all links. The higher the value of the threshold, the less similar the clusters connected by the link are. This threshold thus provides an objective method of determining the number of clusters without heavy reliance on domain specific information. The inconsistency coefficient of the links in the cluster tree can identify cluster divisions where the similarities between data objects change abruptly. A link whose height differs noticeably from the height of the links below it indicates that the clusters at this level in the dendrogram are much farther apart from their child clusters. This link is said to be inconsistent with the links below it. As we move from the top level (root node) towards the lower levels (leaf nodes) the heights of the links become approximately the same height as the links below them, thus indicating that there is no distinct division between the clusters at this particular level in the hierarchy.

Figure 3 depicts a dendrogram with consistent and inconsistent links. It can be seen that the objects in the dendrogram fall into two groups that are connected by links at a much higher level in the tree. These links are inconsistent when compared with the links below them in the hierarchy.

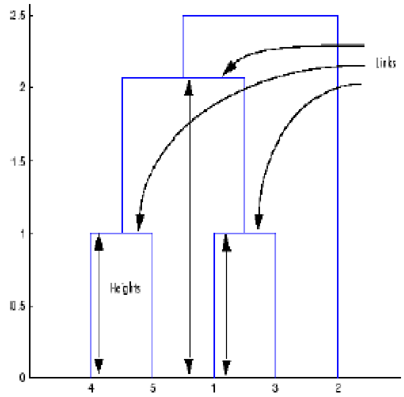
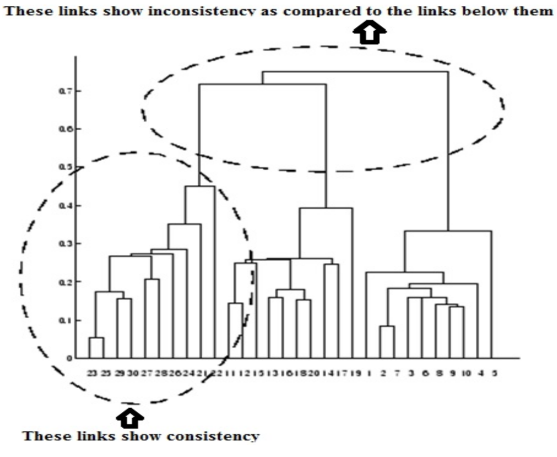


Fig.2. Dendrogram structure indicating links and associated heights


These links show inconsistency as compared to the links below them

These links show consistency

Fig.3. Dendrogram showing consistent and inconsistent links

We take the inconsistency threshold value at such a level in the hierarchy as the cut-off point. After determining the cut-off point, we give each cluster a unique label and extract the clustered data from each level using the procedure shown in lines 2 to 5 of Algorithm 1. Then, we increment the data abstraction level and cluster count in lines 6 and 7. After incrementing the level, we get the similarity value and store it as a spilt point as shown in lines 8 and 9. This value is the actual Euclidian distance where a cluster splits into two child clusters. Line 10 checks the threshold (cut-off point) in the tree. If the split point is less than the threshold value then we recursively call on the

Binary_cluster_tree method for the two left and right child clusters as shown in line 12 and 13. Otherwise, we stop the recursive methods and output the number of clusters and total levels in the cluster hierarchy.

The knowledge contained within a cluster is captured by the relationships that exist between numeric variables and the nominal variables. In general, relationships between nominal and numeric variables are subject to change depending on the range that the numeric variables are constrained on. As the range tightens at the lower levels of the dendrogram, significant differences in the relationships emerge, as shown in the results of the two case studies that we undertake in Section 5.

Algorithm 1. Generate Binary Tree

```
Input:    Node,                  //Node is the root of the tree having complete data in the first call of the method
          TH,                    //Calculated threshold value for cut-off point
          Similarity_value       // similarity value where the root node divides into two clusters
          Level_id   // level of data abstraction

Output:  Num_of_clusters, Level_id
Method: Binary_cluster_tree (Node, Level_id)

           // Initialization of input variables
     1.   Level_id ← 1;  Num_of_clusters ← 1; Spilt_point ← Similarity_value; cluster_label ← ‘C’

          // create two new child nodes and add extract data present in nodes

     2.   Node.Left.Childlabel   = Node.cluster_label + ‘1’       /* left child cluster label is concatenated with integer 1
     3.   Node.Right.Childlabel = Node.cluster_label + ‘2’    /* right child cluster label is concatenated with integer 2
     4.   Node.Left.Child.Data   = [ d ϵ Node.data | d is left child data objects of parent node ]
     5.   Node.Right.Child.Data = [ d ϵ Node.data | d is right child data objects of parent node ]

          // Increment data abstraction level and cluster count

     6.   Level_id ← Level_id + 1
     7.   Num_of_clusters ← Num_of_clusters + 2
     8.   Get Similarity_value at Level_id  /* similarity value indicate the similarity among the objects in a cluster
     9.   Spilt_point ← Similarity value

          // check cut-off point in the tree to stop the recursive method
     10.  If (Spilt_point < TH)
          // recall of binary tree generation method for each parent cluster (node) that splits into two child clusters
                 Binary_cluster_tree (Node.Left.Child, Level_id)
                 Binary_cluster_tree (Node.Right.Child, Level_id)
     else return Num_of_clusters, Level_id
```

Our preference for Agglomerative Hierarchical Clustering (AHC) is due to the fact that it tends to capture a natural hierarchy more faithfully than other clustering approaches (Seo et al., 2003, Seo et al., 2004, Usman et al., 2010, Usman and Pears, 2011a, Usman and Pears, 2010). In hierarchical agglomerative clustering, only numeric variables play an active part in cluster formation, which is in common with many other clustering approaches. With AHC, nominal variables are normally required to be transformed into numeric form in order to be involved in the clustering process. However, our methodology does not require any such mapping and we believe that nominal variables should retain their original data format as ad-hoc and unnecessary mappings could result in loss of information or may lead to erroneous results. In place of ad hoc transformations we rely on the use of entropy and information gain measures to extract natural groupings of nominal variables within a cluster.

We applied the agglomerative hierarchical clustering algorithm on the 3 numeric variables from our exemplary dataset *D* to generate hierarchical clusters at different data abstraction levels. It produced the dendrogram depicted in Figure 4. Using Algorithm 1, we identified and labelled the hierarchical clusters by giving simple abbreviations at different levels of data abstraction in the form of a binary tree as represented in Figure 5.

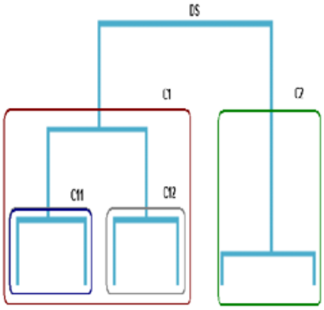


Fig.4. Dendrogram structure of hierarchical clusters (Usman et al., 2013)

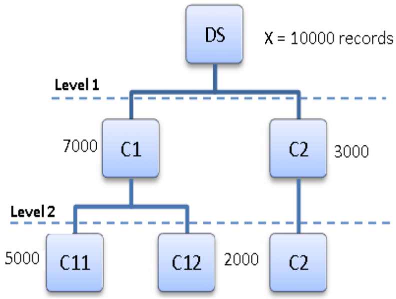


Fig.5. Tree structure of hierarchical clusters (Usman et al., 2013)

4.2. Rank numeric variables using PCA

After the dendrogram is generated, each of the numeric variables within a cluster is ranked by PCA in terms of the degree of variance it captures across the data present in the cluster. PCA is a popularly used statistical technique that has been applied in a wide variety of applications for finding patterns in high dimensional data. As cited by (Uguz, 2011), it can be viewed as a domain independent technique which is applicable to a wide variety of data. The main advantage of PCA is its ability to transform a given set of variables into a new (smaller) set of variables that capture the most variation (Phaladiganon et al., 2013). In this section, we provide an overview of PCA as a method of data reduction.

Suppose that the dataset to be reduced has *n* numeric variables. PCA projects the original dataset onto a smaller dataset that captures most of the variation present in the original dataset. It accomplishes this by finding a set of Eigen vectors $E_{1,}\ldots,E_n$. Given a dataset *D*, we first project the dataset onto its numeric variables and obtain another dataset *D'*. Now from *D'*, the covariance of every pair of items can be expressed in terms of its covariance matrix *M*. The matrix *P* of all possible Eigen vectors can then be derived from $P^{-1}MP = Q$ where *Q* is the diagonal matrix of Eigen values of *M*. Our use of PCA is to obtain a set of factor loadings from the set of Eigen vectors obtained from expression 1 above. In practice, only a subset of Eigen vectors that capture t% of the total variance across dataset *D'* is used. Each Eigen vector E_i is associated with an Eigen value e_i that represents the proportion of variance that is explained by that Eigen vector. The Eigen vectors can then be arranged in ranked order of their Eigen values and the first *m* such vectors that collectively capture at least t% (generally set to 0.95) are chosen for extraction of the factor loadings. The factor loading F_i for an original numeric variable V_i is then given by its communality (Tryfos, 1998). Thus,

$$F_i = \sum_{j=1}^{m} \left(E_{i\,j}\right)^2 \quad \forall\; i = 1,\ldots,n \tag{2}$$

The factor loadings F_i obtained are then used to rank the numeric variables. In order to obtain the ranked list of numeric variables in a parent cluster, say C1, we apply PCA on the numeric variables of the two child clusters, namely C11 and C12 and obtain the factor loadings (Eigen values) for each numeric variable present in these child clusters. We then compare difference between the loadings for each numeric variable across clusters C11 and C12. Each variable is assigned a ranking at a (parent) cluster that is equal to the difference in factor loadings for that variable across the child clusters. The greater the difference in factor loadings is for a given variable, the higher the rank for that variable.

The rationale behind this approach is that the two mutually exclusive child clusters have the necessary information to identify the numeric variables that defined the split. Thus, if Profit, Quantity and Weight defined the split of cluster C1 (parent) into clusters C11 and C12 (children), then the variable that discriminates most between the two clusters would tend to capture a high degree of variation in one of the clusters while expressing itself to a much lesser extent in the other cluster. Thus for example, Profit expresses itself much more strongly in cluster 1 when compared to cluster 2. The variable Profit has the highest difference in factor loadings amongst the 3 variables, thus acquiring the highest rank, followed by Weight and Quantity as shown in Table 1.

Table 1. Ranking of numeric variable in cluster C1

Numeric Variables	C11 Factor Loadings	C12 Factor Loadings	Comparison Results	Rank
Profit	0.825	0.253	0.572	Rank # 1
Quantity	0.634	0.531	0.103	Rank # 3
Weight	0.726	0.512	0.214	Rank # 2

4.3. Rank nominal variables using information gain

In this step, we rank the nominal variables present in each data cluster. To achieve this objective, we adopt entropy and information gain measures. Entropy is a measure that indicates the degree of impurity in a variable. It can be measured in bits for a variable say, *v* through equation 3.

$$\text{Entropy } (v) = - (p * \log (p) + (1-p) * \log (1-p)) \qquad (3)$$

Generally, entropy is greater if the distinct values in a variable are evenly distributed and vice versa. Information gain, on the other hand, is a measure of purity in a variable. The variable with the highest information gain is usually used as a splitting variable and is computed by equation 4 below.

$$\text{Information Gain } (v) = \text{Entropy } (v) \text{ before spilt} - \text{Entropy } (v) \text{ after split} \qquad (4)$$

We calculate the information gain for each nominal variable present in a cluster in order to rank the variable in terms of significance. The variable with the highest information gain acquires the highest rank as it minimizes the information required (i.e. has the least randomness) to cluster the records from the parent cluster, say C1, into child clusters, namely C11 and C12. However, we need to take into account the entropy on the left child cluster (C11) and the right child cluster (C12) in order to calculate the entropy after a parent cluster (C1) splits. Equation 5 defines the entropy of a variable after the spilt.

$$\text{Entropy } (v) \text{ after split} = \text{Wfl} * (\text{Entropy } (v) \text{ left child cluster}) + \text{Wfr} * (\text{Entropy } (v) \text{ right child cluster}) \qquad (5)$$

In equation 5, Wfl and Wfr represents the weight factors, which are the ratios of the number of records on the left child (C11) and the right child (C12) clusters respectively over the number of records in the parent cluster (C1). Therefore, by comparing the entropy before and after the split, we obtain a measure of information gain, or in simple terms, we assess the information that was gained by performing a split with a given variable v. We illustrate this step using our running example having three nominal variables, namely quality, color and size. Our objective is to rank the nominal variables present in the parent cluster C1. We start by first calculating the entropy of each variable present in parent cluster C1 and the two child clusters C11 and C12 using equation 3. The results obtained are shown in Table 2. Thereafter we calculate the entropy of each variable after the split by substituting the values from Table 2 in equation 5. For example, the entropy of variable Quality can be calculated as follows:

$$\begin{aligned} \text{Entropy after split} &= \mathit{Wfl} * (\text{Entropy (Quality) left child}) + \mathit{Wfr} * (\text{Entropy (Quality) right child}) \\ &= 5000/7000 * (2.042) + 2000/7000 * (1.304) \\ &= (0.714 * 2.042) + (0.285 * 1.304) \\ &= 1.457 + 0.371 = 1.828 \end{aligned}$$

Table 2. Calculated entropy of nominal variables

	Cluster C1	Cluster C11	Cluster C12
Variables	entropy (parent)	entropy (left-child)	entropy (right-child)
Quality	2.028	2.042	1.304
Color	1.845	1.936	1.687
Size	1.596	1.600	1.404

We calculated the weight factors for the left and right child clusters by utilizing the distribution of records across clusters given in Figure 2. We then calculated the information gained for the Quality variable using equation 3.

Information Gain = Entropy before spilt – Entropy after split
= 2.028 – 1.828 = **0.2**

Similarly, we calculated the information gain for the other two variables and ranked them accordingly. The results of the ranking for this example are shown in Table 3.

4.4. Apply multidimensional scaling

After ranking the nominal variables based on the information gain measure, the next step of our methodology is to obtain natural groupings for each of the nominal variables. To achieve this, we applied multidimensional scaling to identify the semantic relationships among values in each nominal variable. With multidimensional scaling semantic relationships between multiple nominal variables can easily be visualized through a parallel coordinates.

Table 3. Ranked list of nominal variables using information gain

		Cluster C1		
Variables	**entropy before**	**entropy after**	**information gain**	**Rank**
Quality	2.028	1.828	0.2	Rank # 1
Color	1.845	1.864	-0.01	Rank # 3
Size	1.596	1.544	0.05	Rank # 2

In a parallel coordinate display each nominal variable is represented on a vertical scale. The values are displayed on the scale and the spacing between the values signifies the similarities and differences between the values. Figure 6 depicts the use of this technique for visualizing each of the nominal variables for our running example. Figure 6 reveals that objects with quality ok or bad have a similar underlying distribution for Color and Size in contrast to the objects with quality as good which have different color and size characteristics. Each of these values represents a numeric value on the scale.

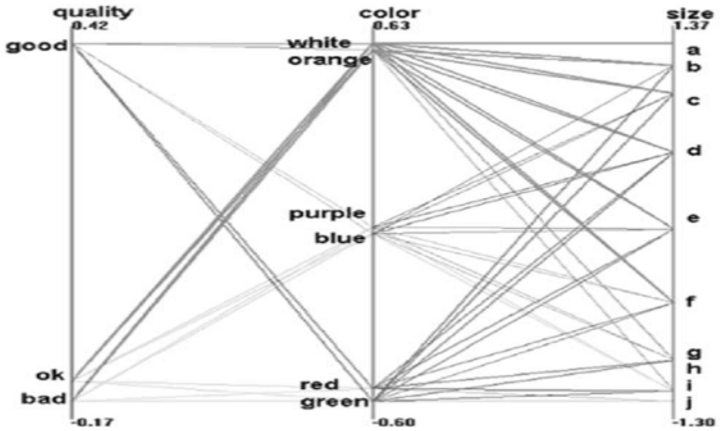


Fig.6. Dendrogram structure of hierarchical clusters (Rosario et al., 2004)

As can be seen from Figure 6 the parallel coordinate display enables the easy grouping of *Color* values. Three natural and distinct groups each having two different sets of colors such as (white, orange), (purple, blue) and (red, green) are clearly defined on the display. However, the Size variable is difficult to group through a simple visual inspection of the parallel coordinate display. There are 10 different sizes starting from *a* to *j* and the distribution of these values on the scale does not permit a precise grouping based on visual inspection. For instance, sizes *a* and *b* are closer to each other but it is difficult to visually determine whether size *c* should be grouped with size *a* and *b* or whether it should be contained within another group with size *d*.

In a real world scenario such variables with large cardinality such as *Country* or *Product codes* are common and it is next to impossible to visualize and group the values by visualization alone. An automated method for grouping is then clearly required. Algorithm 2 generates groups for a given nominal variable given its coordinates produced by the multidimensional scaling method. A generic grouping strategy is used to assign values to groups, where each group corresponds to a collection of semantically related values. We first take the minimum and maximum value of each coordinate and calculate a threshold for assigning values. The threshold is computed as the average range, taken across all nominal values to groups (lines 1 to 3). Nominal values are then assigned to groups on the basis of proximity to each other (lines 5 to 14). If two consecutive values are not further from each other than the threshold distance then they are assigned to the group, otherwise they fall into two neighbouring groups. After generating the groups, we check for singleton groups (lines 15 to 21). If such singleton groups exist, they are merged into a single group called the "outlier" group. After ranking the numeric and nominal variables and obtaining the natural groupings in each of the nominal variables, we move to the next step of creating a multidimensional schema.

Algorithm 2. Grouping similar nominal values

```
Input:    Nom_values // values in each nominal variable
          Nom_values_count // total number of values present in a nominal variable
Output: No_of_groups, //Distinct groups having similar nominal values
Method: Grouping (Nom_values)

          // find minimum and maximum value of a nominal variable and calculate threshold of grouping

1.        Min_value ← min (Nom_values)
2.        Max_value ← max (Nom_values)
3.        Th ← (Max_value – Min_value) / (Nom_values_count – 1)
4.        j ← 1; Group_others [] ← 0;      * Group_others is created to  add the outlier values present in each variable

          // create groups of similar values based on the calculated threshold

5.        Group[j].add(Nom_values[i])                    * the first nominal value is added to the first group
6.        for (i=1; i<Nom_values_count; i++)
7.            diff_btw_values = Nom_values [i + 1] – Nom_values [i] * difference between two consecutive values is calculated
8.            if  (diff_btw_values < Th)
9.              Group[j].add(Nom_values[i+1])
10.           else
11.             Group[j+1].add(Nom_values[i+1])
12.              j ← j + 1
13.              No_of_groups ← j
14.            endif
15.       endfor

          // create outliers group by merging single valued groups together

16.       for (Group=1; Group<No_of_groups; Group++)
17.           Group_value_count ← count (values.Group)          * count the number of values present in a created group
18.           if (group_value_count = = 1)
19.             Group_others ← add.value.Group
20.             j ← j -1                        * decrement the total number of groups when we delete single valued groups
21.           else
22.              No_of_groups ← j
23            endif
24.         endfor
25.     return No_of_groups
```

4.5. Create multidimensional schema

After receiving ranked lists of numeric and nominal variables, a multidimensional *STAR* schema is created by treating nominal variables as dimensions and numeric variables as facts. We produce the schema with all dimensions and facts present in a data cluster. The groupings information assists in defining the dimensional hierarchy or dimensional levels. Each dimension in a cluster has a group level and value level. For example, if Color is a dimension then it has Color (All) level → Color_groups (Group) level → Color_names (Value) level. A physical structure is created with the use of generic *SQL* queries. These queries create the necessary tables (fact and dimension) and define table relationships that are needed to implement the multidimensional schema. These generic queries are automatically structured to support the quick generation of multidimensional schema for any given cluster in the hierarchical tree.

For our running example, we construct the multidimensional schema by taking the nominal variables as dimensions and numeric variables as facts. Figure 7 depicts the multidimensional schema for cluster C1. In this step, the schema contains all the dimensions and facts present in data cluster C1. At this stage the user has the option to specify the top *k* and *m* thresholds, where *k* is the number of the highest ranked dimensions and *m* is the number of highly ranked facts to be selected for data cube construction. The multidimensional schema is used to construct informative data cubes in the next step.

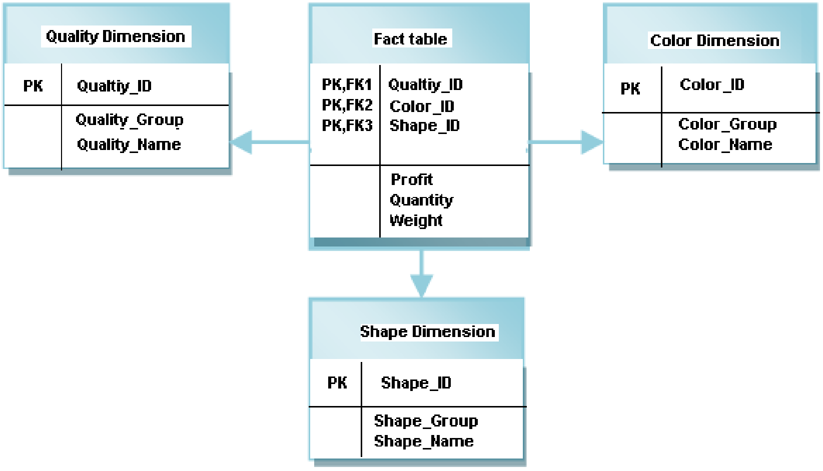


Fig.7. Multidimensional schema of cluster C1 (Usman et al., 2013)

4.6. Construct informative data cubes

In this step, a data cube is constructed using the highly ranked dimensions and facts present in the generated multidimensional schema. The user can constrain the search space by selecting either the top ranked dimensions/facts or the dimensions/facts of his/her own choice from the generated schema. The purpose of providing this flexibility is that we do not want to restrict the user only to the exploration of statistically significant (highly ranked) dimensions/facts in cases where specialized domain knowledge can contribute to the selection of significant facts and dimensions from an application perspective.

We believe each user has specific data analysis requirements and there may be certain cases that the user would like to see a highly ranked dimension with low ranked facts or vice versa. The construction of informative data cubes allows the user to apply basic OLAP operations such as Drill-down, Roll-up, Slice and Dice in order to interactively explore the data to find meaningful patterns. However, pattern discovery through the use of OLAP is limited by the analyst's insights. Such insights may not extend to patterns that are hidden due to the sheer data volume and dimensionality of the data. Furthermore, data granularity introduces another complicating factor: patterns themselves change depending on the level of granularity, as we proceed down the dendrogram the dynamics of relationships between variables are likely to change. For these reasons we apply association rule mining to enhance the knowledge discovery process.

4.7. Mine association rules from schema

In this step, we apply the well-known *Aprori* algorithm (Agrawal et al., 1993) in order to generate rules from the multidimensional schema. Association rule mining is a widely is a well-recognized data mining procedure (Nahar et al., 2013) in which a rule is defined as an implication A → B where A and B, the disjoint sets, are the frequent items in the data. Strong rules meet user-specified thresholds known as minimum support ***s*** and minimum confidence ***c***. Support reflects the percentage of records that contain both A and B, while Confidence refers to the percentage of records containing B that also contain A. Both these measures are used to specify the significance of a rule.

Besides the advantages of easy implementation and the usage of large itemset property, *Aprori* has a number of inefficiencies and limitations. For instance, it requires several iterations of the data and produce a large number of meaningless redundant rules. Moreover, it only works on nominal attributes and the numeric attributes need to be discretized. However, we would like to clarify at this point that we have only utilized *Aprori* in order to demonstrate rule mining step of our methodology. Our main contribution lies in the proposed methodological framework for discovery diverse rules. *Aprori* can easily be replaced entirely by any other more efficient or effective algorithm without affecting the methodological framework proposed in this paper. For instance, in datasets with large number of frequent itemsets, weighted association rule mining algorithm proposed by (Pears et al., 2013) could possibly be used in order to efficiently find important rules by assigning automatic weights to important items in large transactional dataset.

In this rule mining step, we utilize the highly ranked dimensions to discover significant associations between them. For instance, in a 4 dimensional data cube, the two dimensions with the highest degree of impurity (largest entropy) can be targeted for rule discovery. Dimensions with high impurity are not easily predictable as the underlying data distribution tends to be uniform. Furthermore, the larger the entropy of a variable, the less information we know about the underlying variable as the underlying data distribution tends to be random in nature. On the other hand, low entropy means that the distribution is less random; the variable may have many low values and a few extreme values. Hence, such variables tend to be more predictable, which means that association rules that contain high entropy variables on the right hand will in general provide more insights than their low entropy counterparts, provided they meet standard rule evaluation thresholds such as rule Confidence and/or Importance.

However, association rule mining has the potential to generate a large number of trivial rules. Although the rule base can be pruned by setting the rule *support* and *confidence* thresholds appropriately, there is no guarantee that the rules that survive would capture interesting patterns in the multidimensional data. One reason is that the support and confidence measures were designed to evaluate rules derived from transactional data and are not necessarily effective for multidimensional data (Nestorov and Jukic, 2003a). This limitation motivated us to introduce the next step of the methodology that is designed to evaluate the interestingness of rules generated from multidimensional schema.

4.8. Mine association rules from schema

In this step, we evaluate the interestingness of the generated rules with the help of alternative evaluation measures. One such measure is known as *Importance* which captures the usefulness of a rule. Rule importance is defined in equation 6.

$$\text{Importance } (A \rightarrow B) = \log(\text{probability } (B|A) / \text{probability } (B|\text{ not } A)) \tag{6}$$

The importance measure assesses the degree of correlation between dimensions. For instance, if the importance of a rule is greater than 0 then it means the dimensions are positively correlated and vice versa. A positive importance means that the probability of observing the right hand side of the rule increases when the left hand side is true. Table 4 shows a list of hypothetical rules generated from cluster C1. We contrast the rules generated from cluster C1 with those generated from the same cluster that has had the multidimensional schema structure imposed on it.

Table 4. Hypothetical rules from cluster C1

Without multidimensional schema			**With multidimensional schema**		
Rule #	**Rules**	**Importance**	**Rule #**	**Rules**	**Importance**
1	If Quality = [Good] and Size [a] → Color = white	1.80	**1**	If Quality [G1] and Size [G2] → Color = white	1.80
2	If Quality [ok] and Size [b] → Color = orange	1.29	**2**	If Quality [G2] and Size [G4] → Color = orange	1.29
3	If Quality [ok] and Size [e] → Color = purple	1.10	**3**	If Quality [G2] and Size [G3] → Color = purple	1.10
4	If Quality [bad] and Size [f] → Color = green	1.05	**4**	If Quality [G2] and Size [G3] → Color = green	1.05

Such rules help in understanding the underlying association among different dimensions. For instance, Rule 1 without the schema predicts that the Color of a product will be “white” if Quality = good and Size = a. Also, the positive importance value indicates that there is a strong relationship between the Quality and Size dimensions. On the other hand, Rule 1 with the use of the schema predicts the same color by giving an association among a group of values present in each dimension, namely G1 and G2. Each group contains a set of diverse values which are semantically related. For instance, Rule 1 with schema, predicts the white color product if it’s Quality belongs to the values of the 1st group G1 = [ok, bad] and its size belongs to the values of 2nd group G2 = [*a*, *b* and *c*]. It is apparent that the importance value of rules generated with and without schema is the same but the rules generated from schema are more diverse in comparison to the rules without schema. For instance, rule 1 with schema predicts the same color but is predicated on diverse Quality and Size values present in semantically related groups. Intuitively, a rule R which has more triggering conditions in the rule antecedent for any given rule consequent is more valuable than a rule S with fewer trigger conditions as rule R is fired in a greater diversity of situations than rule S. However, we cannot conclude merely on the basis of multiple values present in the dimensional groups that the rules generated from the schema are more diverse. We need further concrete evidence to support our claim.

In order to do that, we conducted further advanced evaluation using the objective measures of interest under diversity criteria, namely Rae, CON and Hill proposed by Zbidi et. al in (Zbidi et al., 2006). These measures provide concrete statistical evidence that the diversity of the set of rules produced from our semi-automatically generated multidimensional schema is higher when compared to the rules generated without schema.

We employ the above mentioned diversity measures for the advanced evaluation of summary tables generated for evaluating rules. These summary tables are basically deduced from the main dataset according to a given rule. For instance, given a rule R1 = Quality [Good] and Size [b] → Color = [white] generated from our example dataset ***D***, the summary table S1 for this rule is a table with the set of records containing Quality = (good) with size = (b). Using these summary tables we evaluate the interestingness of rules. The three measures Rae, CON and Hill are defined in equations 7, 8 and 9 respectively.

$$Rae = \sum_{i=0}^{m} \frac{ni\left(ni-1\right)}{N\left(N-1\right)} \qquad (7) \qquad CON = \sqrt{\frac{\left(\sum_{i=1}^{m} Pi^{2}\right) - \bar{q}}{1-\bar{q}}} \qquad (8) \qquad Hill = 1 - \frac{1}{\sqrt{\sum_{i=1}^{m} Pi^{3}}} \qquad (9)$$

Here, m denotes the total number of rows in a summary table; ni is the value of derived count attribute of each row in the summary table; N is the total count $\mathrm{N} = \sum_{i=1}^{m} ni$; $Pi = \frac{ni}{N}$ is the actual probability of row ri ; $\bar{\mathrm{q}} = \frac{1}{m}$ is the uniform probability of row ri. With the help of these measures we rank the generated rules in terms of the diversity of information contained in each rule. Table 5 depicts the values obtained for the diversity measures on the rules that were generated.

Table 5. Rule evaluation using advanced diversity measures for cluster C1

Rule set	Without multidimensional schema			With multidimensional schema		
	Rae	CON	Hill	Rae	CON	Hill
R1-R2	0.53	0.56	-2.7	**0.84**	**0.87**	**-0.3**
R3-R4	0.25	0.36	-3.7	**0.53**	**0.58**	**-1.7**

It is apparent from Table 5 that all three advanced diversity measures show a significant improvement for the rules generated with the multidimensional schema. We now turn our attention to the application of the methodology on two real-world datasets.

5. Case study results on real world datasets

In order to validate our proposed methodology, we have adopted the case study research method in order to demonstrate the effectiveness of our proposed methodology. It is a very popular method with practitioner researchers and it allows the use of experiments which meets the needs of our research. This case study approach allowed us to study each of the three cases on real world datasets in depth in order to formulate a solution, which is the systematic methodology for discovering diverse association rules.

We conducted three case studies on real world datasets, namely Automobile (Schlimmer, 1985), Adult (Kohavi and Becker, 1996) and CoverType (Blackard et al., 1998) taken from the UCI machine learning repository (Asuncion and Newman, 2010). These three datasets feature a rich mix of numeric and nominal variables and are thus ideal to illustrate the implementation of our proposed methodology.

5.1. Case study 1 – Automobile dataset

We chose the *Automobile* dataset having 26 variables for our first case study. It has 205 records with 11 nominal and 15 numeric variables. This benchmark dataset describes the specification of an automobile in terms of various characteristics, its assigned insurance risk rating and its normalized (financial) losses in use as compared to other cars. More details of this dataset can be found at the University of California - machine learning website.

As per our first step of the proposed methodology, we applied agglomerative hierarchical clustering using the Hierarchical Clustering Explorer (HCE) tool developed by Jinwook and Shneiderman (Jinwook and Shneiderman, 2002) to generate a dendrogram. From the dendrogram generated we determine the suitable cut-off point and generate the binary tree using the procedure explained in Algorithm 1. The calculated threshold for the cut-off point was 0.676 so we cut the dendrogram at this value and considered it to be the last level of our data abstraction. There were a total of 5 levels of data abstraction until the cut-off point and the last level had 10 clusters in it.

To implement the second step of our proposed methodology, we used IBM’s SPSS software to apply PCA analysis on each cluster. Firstly, we plotted the 16 numeric variables present in each cluster and identified that most of the clusters were discriminating well on only 1 component or factor as shown in Figure 8.

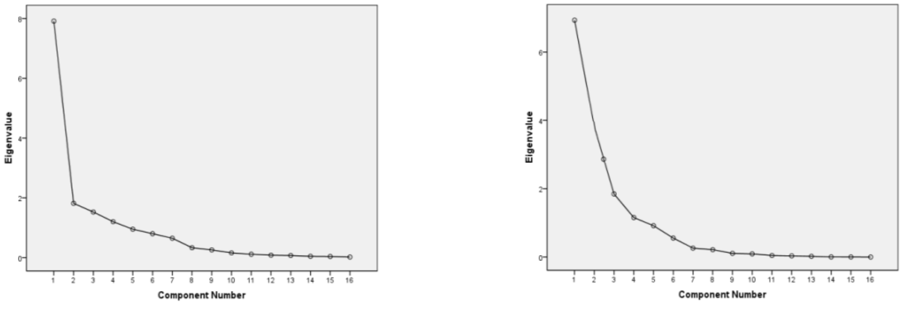
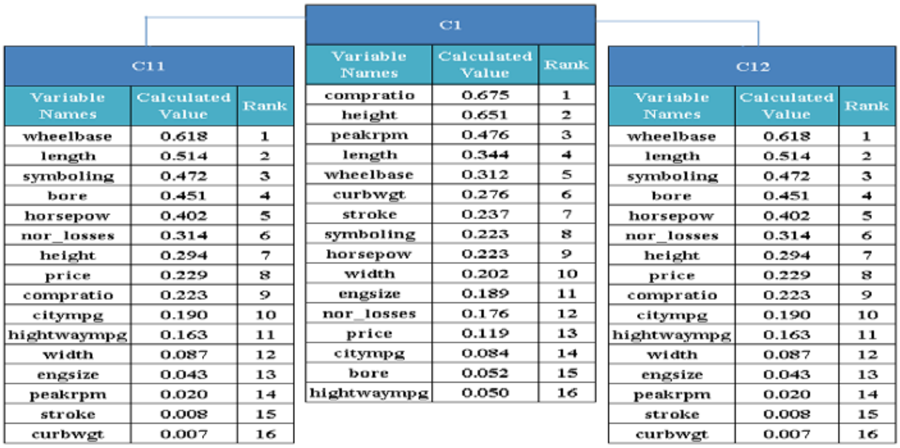


Fig.8. Scree plot showing Eigen values of cluster C11 and C12 (Usman et al., 2013)

The number of components to be extracted is based on Eigen value analysis of all 16 numeric variables present in each cluster. The cut-off point for component extraction can be set by the user to a certain percentage of variance captured by a variable, for example 80%, 85%, 90% or 95%. As explained in the example presented in Section 4, we compared the factor loadings of two child clusters to rank the numeric variables in the parent cluster. Figure 9 shows the ranking of all 16 numeric variables in three clusters namely C1, C11 and C12 after performing PCA and comparative analysis as explained in section 4.2.



C1

Variable Names	Calculated Value	Rank
compratio	0.675	1
height	0.651	2
peakrpm	0.476	3
length	0.344	4
wheelbase	0.312	5
curbwgt	0.276	6
stroke	0.237	7
symboling	0.223	8
horsepow	0.223	9
width	0.202	10
engsize	0.189	11
nor_losses	0.176	12
price	0.119	13
citympg	0.084	14
bore	0.052	15
hightwaympg	0.050	16

C11

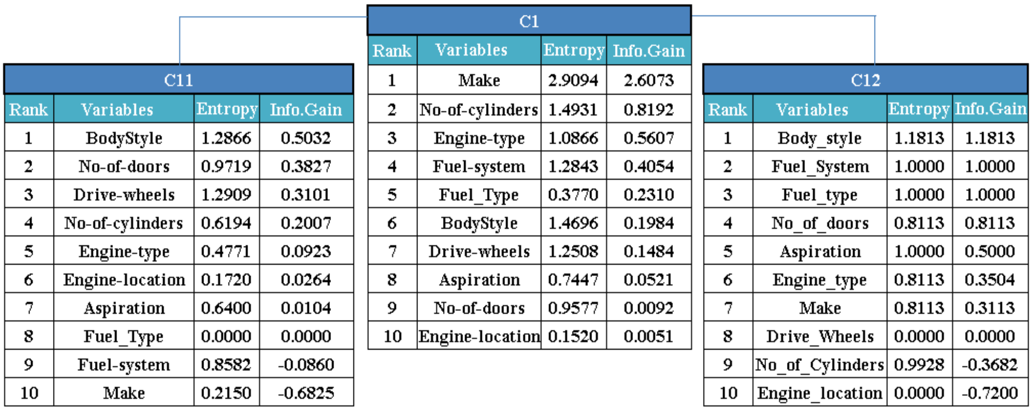
Variable Names	Calculated Value	Rank
wheelbase	0.618	1
length	0.514	2
symboling	0.472	3
bore	0.451	4
horsepow	0.402	5
nor_losses	0.314	6
height	0.294	7
price	0.229	8
compratio	0.223	9
citympg	0.190	10
hightwaympg	0.163	11
width	0.087	12
engsize	0.043	13
peakrpm	0.020	14
stroke	0.008	15
curbwgt	0.007	16

C12

Variable Names	Calculated Value	Rank
wheelbase	0.618	1
length	0.514	2
symboling	0.472	3
bore	0.451	4
horsepow	0.402	5
nor_losses	0.314	6
height	0.294	7
price	0.229	8
compratio	0.223	9
citympg	0.190	10
hightwaympg	0.163	11
width	0.087	12
engsize	0.043	13
peakrpm	0.020	14
stroke	0.008	15
curbwgt	0.007	16

Fig.9. Ranking of numeric variables present in *Automobile* dataset

It is clear from Figure 9 that the ranking of numeric variables differ greatly between clusters. In other words, each data cluster has its own unique set of significant numeric variables. For instance, *compratio (Compression Ratio)*, *height* and *peakrpm (Peak revolutions per minute)*, which happen to be the most significant variables in cluster C1, do not appear to be significant at a lower level of data abstraction. Interestingly, the most significant variable *compratio (Compression ratio)* in C1 has a low rank of 9 in child clusters C11 and C12. Similarly, *height and peak-rpm* moved from 2nd and 3rd places to 7th and 14th places respectively in clusters C11 and C12. Thus we can infer that the automobiles in cluster C1 (parent) are split into two separate clusters primarily on the basis of *comp-ratio* and *height* variables and marginally on the basis of *bore* and *highwaympg (Highway miles per gallon)* variables. After establishing the significant numeric variables in each cluster, we rank the nominal variables in each cluster based on information gain measure as described in Section 4.3 of the paper. Figure 10 shows the information gain based ranking for three clusters, namely C1, C11 and C12.



C1

Rank	Variables	Entropy	Info.Gain
1	Make	2.9094	2.6073
2	No-of-cylinders	1.4931	0.8192
3	Engine-type	1.0866	0.5607
4	Fuel-system	1.2843	0.4054
5	Fuel_Type	0.3770	0.2310
6	BodyStyle	1.4696	0.1984
7	Drive-wheels	1.2508	0.1484
8	Aspiration	0.7447	0.0521
9	No-of-doors	0.9577	0.0092
10	Engine-location	0.1520	0.0051

C11

Rank	Variables	Entropy	Info.Gain
1	BodyStyle	1.2866	0.5032
2	No-of-doors	0.9719	0.3827
3	Drive-wheels	1.2909	0.3101
4	No-of-cylinders	0.6194	0.2007
5	Engine-type	0.4771	0.0923
6	Engine-location	0.1720	0.0264
7	Aspiration	0.6400	0.0104
8	Fuel_Type	0.0000	0.0000
9	Fuel-system	0.8582	-0.0860
10	Make	0.2150	-0.6825

C12

Rank	Variables	Entropy	Info.Gain
1	Body_style	1.1813	1.1813
2	Fuel_System	1.0000	1.0000
3	Fuel_type	1.0000	1.0000
4	No_of_doors	0.8113	0.8113
5	Aspiration	1.0000	0.5000
6	Engine_type	0.8113	0.3504
7	Make	0.8113	0.3113
8	Drive_Wheels	0.0000	0.0000
9	No_of_Cylinders	0.9928	-0.3682
10	Engine_location	0.0000	-0.7200

Fig.10. Ranking of nominal variables using information gain measure

Similar to numeric variables, the list of ranked nominal variables also shows sharp differences as we move from a higher level of data abstraction to a lower level. It can be seen from Figure 10 that *Make*, which is a top ranked in cluster 1, is lowly ranked in cluster C11. Similarly, *No-of-Cylinders*, which is the second most significant variable, took 9th place in cluster C12. In other words, *Make* and *No-of-Cylinders* had the least randomness in parent cluster C1 but at an immediate lower level in the hierarchy these variables appear to have the most randomness or impurity. We suggest that these highly ranked variables should be explored as a first choice in a given cluster C1 as they possess more versatile information that defines the split as compared to the lowly ranked variables which play a minimum role in the cluster split.

After ranking the nominal variables, we applied multidimensional scaling and grouped the values present in each nominal variable using Algorithm 2. Figure 11 shows the groupings obtained for the top ranked variable (Body-Style) in cluster C11 and C12.

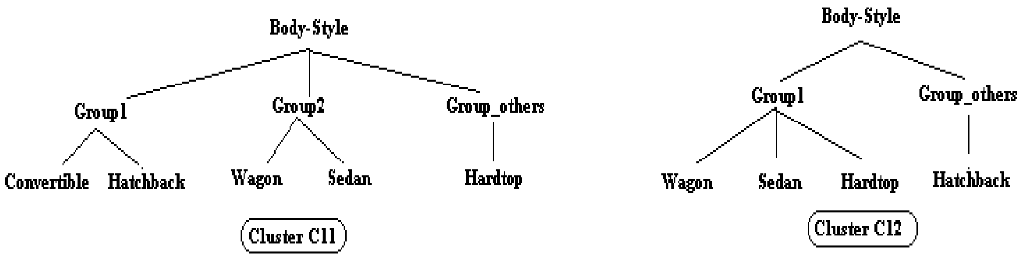


Fig.11. Grouping obtained after multidimensional scaling and application of Algorithm 2

We see that there is a difference in the number of groups and the values present in the groups across the two child clusters for the same nominal variable. Interestingly, *Hardtop*, which had no similarities with any other body-style type in cluster C11, shows an affinity with *Sedan* and *Wagon* types in the sibling cluster C12. Moreover, *Hatchback* type becomes an outlier (having no similarities with any other body-style type) in cluster C12. Such unique and sharply contrasting patterns are difficult to obtain by relying on manual exploration alone in high dimensional and high volume datasets.

The ranked lists of numeric and nominal variables serve as a starting point to constrain the multidimensional space. At this point the appropriate number of important dimensions (nominal variables) and facts (numeric variables) are selected for the creation of multidimensional schema. After creating the multidimensional schema we construct data cubes at different levels in the cluster hierarchy. Figure 12 shows the 3 dimensional cubes structure of two clusters (C1 and C11) at different levels of the hierarchy.

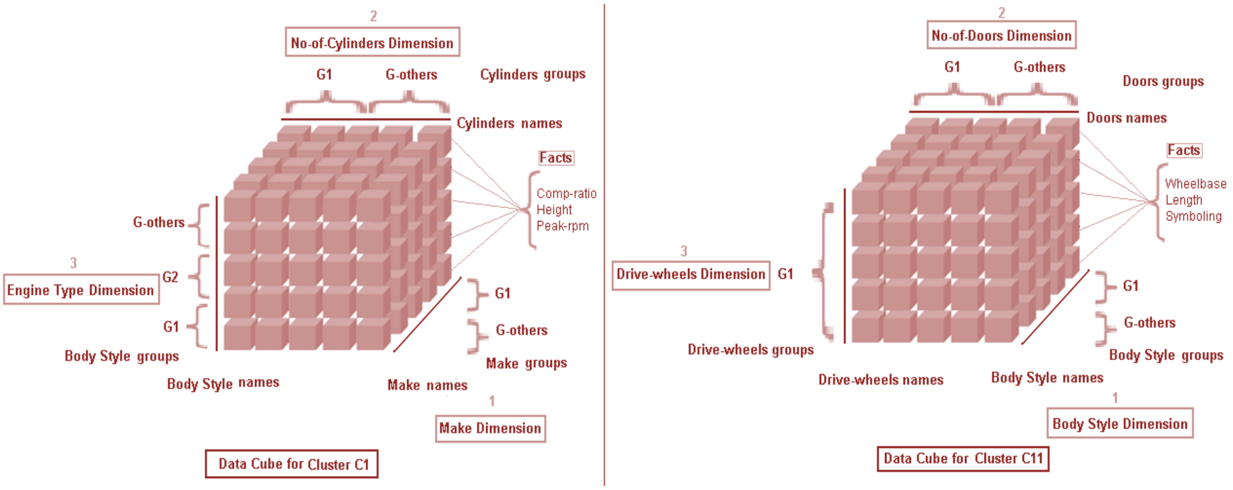


Fig.12. Comparison of 3-dimensional cubes at different levels of hierarchy

It can be seen from Figure 12 that the top 3 dimensions and facts in data cube C1 are totally absent in data cube C11. Our methodology suggests unique combinations of dimensions and facts for cube exploration using OLAP analysis.

For instance, in data cube C1, the three suggested facts namely, *Comp-ratio, Height*, and *Peak-rpm* when explored through the ranked dimensions (*Make, No-of-Cylinders* and *Engine-type)* would give more significant information as compared to the lowly ranked dimensions and facts. Consider data cube C1 as an example; the three facts (*Comp-ratio, Height* and *Peak-rpm)* can be explored through 22 different *Make* types, 7 different *Engine* types and 6 distinct values for *No-of-Cylinders.* This makes a total of (22 x 6 x 7) 924 data cells in the cube to be explored through standard OLAP analysis. By grouping together similar dimensional values this search space can be reduced significantly, while enabling meaningful patterns to be discovered.

After grouping, we see that there are only 2 groups for *Make,* 3 groups for *Engine-type* and 2 groups for *No-of-Cylinders* dimensions, thus yielding only (2x3x2) 12 distinct areas in the data cube for exploration. The grouping not only reduces the search space but also provides groups of dimensional values that have intra-group semantic affinity with each other. Furthermore, outliers for each dimension are highlighted by inserting values that are distant from all others in the *Group-others.* For instance, in the *Make* dimension, Subaru and Porsche are the two types of automobiles grouped in *Group-others* which bear no relation to the other 20 automobiles that were grouped together in *Group1.*

However, OLAP analysis by itself is limited in finding the associations and correlations that could exist within dimensions. Although our proposed methodology provides a more constrained space for the OLAP user to find patterns, pattern discovery could still be a laborious task. A case can be made that even in the constrained space a user may require some intelligent assistance in order to find hidden associations among the dimensions. On the other hand, after finding a behavioural pattern through OLAP analysis, an OLAP user may require further support to drill down to find the dimensional attributes influencing that pattern of interest. In order to deal effectively with such cases, we apply the *Apriori* algorithm for discovering multidimensional rules.

In order to assess the benefits of the multidimensional schema on knowledge discovery we apply rule mining on multidimensional schema for three clusters C1, C11 and C12 and compare the rule bases generated with those obtained from the original cluster data. Table 6 shows the top 10 rules generated for cluster C1 at a minimum probability value of 0.4 and minimum importance value of 0.10. Based on these thresholds the algorithm produced 59 rules in total. From the rule base produced we see that the two most highly ranked dimensions (*Make* and *No-of-cylinders)* have a strong tendency to predict the value of the low ranked dimension (*Body-Style).* Basically, the rules indicate that certain makes of cars are distinctive in terms of their body style as some of the Alpha-Romeo models are convertibles whereas the Porsche models tend to come in hardtop versions.

Table 6. Rules generated from cluster C1 flat data

RULES WITHOUT SCHEMA (CLUSTER C1)		
No	**Rules**	**Importance**
R1	Make = alfa-romeo, No Of Cylinders = four → Body Style = convertible	1.410
R2	Make = alfa-romeo, No Of Cylinders = [*All*] → Body Style = convertible	1.310
R3	Make = porsche, No Of Cylinders = six → Body Style = hardtop	1.134
R4	Make = porsche, No Of Cylinders = [*All*] → Body Style = hardtop	0.981
R5	Make = plymouth, No Of Cylinders = four → Body Style = wagon	0.493
R6	Make = renault, No Of Cylinders = four → Body Style = wagon	0.493
R7	Make = dodge, No Of Cylinders = four → Body Style = wagon	0.493
R8	Make = volkswagen, No Of Cylinders = four → Body Style = wagon	0.493
R9	Make = mazda, No Of Cylinders = two → Body Style = hatchback	0.483
R10	Make = alfa-romeo, No Of Cylinders = six → Body Style = hatchback	0.362

Table 7 shows the rules generated with the multidimensional structure imposed on cluster C1. We note that while the rules generated are more informative, the *Make* and *No Of Cylinders* terms now refer to groups rather than individual values. The dimensional schema has produced rules that distinguish classes of automobiles from each other. One class belongs to those automobiles whose makes are in group 1, who have x, y and z number of cylinders and are *convertibles*. On the other hand, when automobiles in another *Make* group (group others) are considered, the body style tends to be *hardtop* instead of *convertible*. These rules are more compact, easier to understand and convey more information to an end user than a plethora of rules that cover each and every combination of *Make* and number of *Cylinders.*

Table 7. Rules generated from multidimensional schema of cluster C1

	RULES FROM SCHEMA (CLUSTER C1)	
No	**Rules**	**Importance**
R1	Make Group = group1, No Of Cylinders Group = group1 →Body Style Name = convertible	1.410
R2	Make Group = group1, No Of Cylinders Group = [*All*] →Body Style Name = convertible	1.310
R3	Make Group = group_others, No Of Cylinders Group = group1 →Body Style Name = hardtop	1.134
R4	Make Group = group_others, No Of Cylinders Group = [*All*] →Body Style Name = hardtop	0.981
R5	Make Group = group1, No Of Cylinders Group = group1 →Body Style Name = wagon	0.493
R6	Make Group = group_others, No Of Cylinders Group = [*All*] → Body Style Name = wagon	0.493
R7	Make Group = group1, No Of Cylinders Group = group_others → Body Style Name = hatchback	0.483
R8	Make Group = group1, No Of Cylinders Group = [*All*] → Body Style Name = hatchback	0.403
R9	Make Group = group_others, No Of Cylinders Group = group1→ Body Style Name = hatchback	0.362
R10	Make Group = group1, No Of Cylinders Group = [*All*] → Body Style Name = sedan	0.302

For instance, the first rule generated without schema can be interpreted as: if the *Make* of an automobile is [*alfa-romeo*] and the *No-of-Cylinders* are [*four*] then the *Body-Style* of that automobile is predicted to be [*convertible*]. On the other hand, the first rule from the schema that utilized the dimensional group level suggests that if *Make* and *No-of-Cylinders* of an automobile belongs to [group1] then the *Body-Style* value is [*convertible*]. Here, [group1] of *Make* has 19 distinct values (*peugeot, jaguar, nissan, mercedez-benz, saab, mazda, toyota, volvo, honda, alfa-romeo, audi, volkswagon, mitsubishi, isuzu, dodge, plymouth, bmw, mercury, renault)* while [group1] of *No-of-Cylinders* has 5 distinct values *(six, twelve, eight, four, five)*. In other words, Rule1 in Table 7 provides much more diverse information as compared to the first rule of Table 6 for the same *importance* value.

In order to quantitatively verify whether the rules produced through our multidimensional design scheme are superior, we evaluated the rule bases on the three objective interestingness measures, *Rae, CON* and *Hill* that were introduced in Section 4.8. For a given cluster we ranked the rules in descending order of rule Importance and then compared the first *k* (where *k* ranges from 6 to 10, in steps of 1) rules produced against each other on the 3 chosen interestingness measures. Table 8 reveals that the multidimensional schema consistently outperforms the raw cluster structure, irrespective of the value of k and the level of data granularity. These results thus confirm through our qualitative analysis that the use of the multidimensional results in the production of more informative and diverse knowledge to the user in the form of rules.

Table 8. Rules interestingness comparison using multiple diversity measures

Cluster labels	**Rule set**	**NO Schema Rae**	**With Schema Rae**	**NO Schema CON**	**With Schema CON**	**NO Schema Hill**	**With Schema Hill**
C1	**R1-R6**	0.139	**0.196**	0.168	**0.221**	-3.919	**-3.473**
	R1-R7	0.116	**0.179**	0.154	**0.233**	-4.701	**-3.823**
	R1-R8	0.100	**0.151**	0.142	**0.203**	-5.502	**-4.682**
	R1-R9	0.086	**0.142**	0.132	**0.195**	-6.321	**-5.446**
	R1-R10	0.076	**0.117**	0.122	**0.167**	-7.154	**-6.287**
C11	**R1-R6**	0.246	**0.291**	0.310	**0.396**	-2.784	**-2.079**
	R1-R7	0.206	**0.271**	0.280	**0.393**	-3.454	**-2.272**
	R1-R8	0.173	**0.220**	0.244	**0.473**	-4.270	**-3.126**
	R1-R9	0.152	**0.189**	0.224	**0.303**	-4.927	**-3.755**
	R1-R10	0.151	**0.186**	0.246	**0.317**	-4.965	**-3.791**
C12	**R1-R6**	0.178	**0.203**	0.196	**0.244**	-3.796	**-3.261**
	R1-R7	0.169	**0.186**	0.233	**0.256**	-3.976	**-3.574**
	R1-R8	0.161	**0.181**	0.253	**0.287**	-4.158	**-3.682**
	R1-R9	0.153	**0.174**	0.265	**0.294**	-4.343	**-3.791**
	R1-R10	0.146	**0.170**	0.271	**0.301**	-4.529	**-3.899**

Finally, we compared the predictive power of rules generated without multidimensional schema with the rules generated from schema. We took *Make* and *No_of_Cylinder* as the input variables in order to predict the *Body_Style* of automobiles. We generated the rules using a randomly chosen subset of 70% of data and then used the remaining 30% to evaluate the predictive accuracy. We ran 10 tests where each test randomly picks unique test data to ensure the predictive accuracy of the generated rules to predict the *Body_Style* of the automobiles accurately. Table 9 shows the accuracy percentage for the 10 tests. It is clear that on average the prediction accuracy of the rules generated from the schema is higher when compared to the rules generated without the use of the schema.

Table 9. Rule prediction accuracy for rules with and without schema

Prediction Tests	Cluster C1		Cluster C11		Cluster C12	
	Schema	**No Schema**	**Schema**	**No Schema**	**Schema**	**No Schema**
Test 1	48.7	43.9	91.4	88.5	100	100
Test 2	48.7	36.5	94.2	85.7	100	100
Test 3	46.3	39	88.5	85.7	100	83.3
Test 4	36.5	41.4	88.5	91.4	83.3	100
Test 5	34.1	43.9	94.2	77.1	100	83.3
Test 6	48.7	43.9	88.5	91.4	100	87.7
Test 7	40.3	33.5	91.2	83.7	83.3	83.3
Test 8	44.7	41	88.5	85.7	100	83.3
Test 9	34.5	36.4	93.5	88.3	83.3	100
Test 10	37.1	31.9	94.2	77.1	83.3	83.7
Average Percentage	**41.96**	39.14	**91.27**	85.46	**93.32**	90.46

From this case study, we note that rule mining performed on the multidimensional schema designed and constructed with the help of hierarchical clustering and multidimensional scaling technique generates diverse rules with greater prediction accuracy.

5.2. Case study 2 – Adult dataset

The second case study was performed on the larger *Adult* dataset. It consists of 48,842 records with 8 nominal and 5 numeric variables as shown in Table 9. This benchmark dataset was extracted from the US Census bureau website using a data extraction system. More details of this dataset can be found at the UCI machine learning website (Asuncion and Newman, 2010). For the first step of hierarchical clustering, we removed missing values from the dataset and used 30,162 records (61% of total) to generate hierarchical clusters.

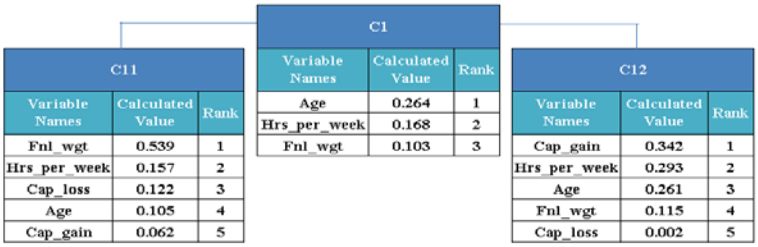
Table 10. Numeric and nominal variables present in *Adult* dataset

Numeric Variables	Nominal Variables	Distinct values in each nominal variable
Age	Sex	2
Hours Per Week	Race	5
Capital Gain	Relationship	6
Capital Loss	Marital Status	7
Final Weight	Work Class	8
	Occupation	14
	Education	16
	Country	41

From the dendrogram generated we calculated the inconsistency coefficient value for determining the cut-off point. The calculated value was 0.623 and so we cut the dendrogram at this value and considered it to be the last level of our data abstraction. There were a total of 9 levels of data abstraction until the cut-off point and the last level had 18 clusters in it. We generated a binary tree of clusters using Algorithm 1.

After generating the hierarchical clusters, we applied PCA for ranking numeric variables and information gain for ranking nominal variables present in each data cluster. The application procedure remained the same as that used for the previous case study. The rankings obtained for this dataset for the first few levels of the hierarchy are shown in Figures 13 and 14.

Similar to the previous case study, the clusters show unique rank lists for both numeric and nominal variables. However, an important point to mention at this stage is the reduction of a few numeric variables from cluster C1. The reason is that 2 variables out of 5 in cluster C1 have variance equal to zero; thereby the Eigen values for those variables having zero variance cannot be calculated. We leave such variables out because they do not play any role in multidimensional analysis. It is apparent that each cluster has a unique ranked list. For instance, *Country* which is the most highly ranked dimension in cluster C1 is the lowest ranked in cluster C11 at a lower level of data abstraction. Similar to our previous case study, we observe the emergence of sharp patterns in both numeric and nominal variables as we go down in the cluster hierarchy.



C1

Variable Names	Calculated Value	Rank
Age	0.264	1
Hrs_per_week	0.168	2
Fnl_wgt	0.103	3

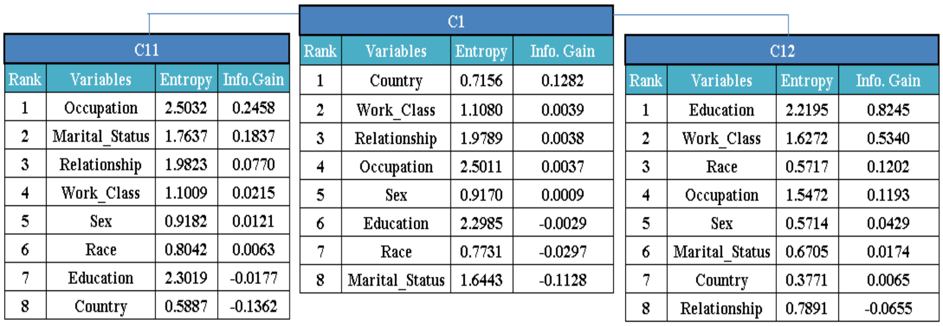
C11

Variable Names	Calculated Value	Rank
Fnl_wgt	0.539	1
Hrs_per_week	0.157	2
Cap_loss	0.122	3
Age	0.105	4
Cap_gain	0.062	5

C12

Variable Names	Calculated Value	Rank
Cap_gain	0.342	1
Hrs_per_week	0.293	2
Age	0.261	3
Fnl_wgt	0.115	4
Cap_loss	0.002	5

Fig.13. Ranked lists of numeric variables present in Adult dataset clusters



C1

Rank	Variables	Entropy	Info. Gain
1	Country	0.7156	0.1282
2	Work_Class	1.1080	0.0039
3	Relationship	1.9789	0.0038
4	Occupation	2.5011	0.0037
5	Sex	0.9170	0.0009
6	Education	2.2985	-0.0029
7	Race	0.7731	-0.0297
8	Marital_Status	1.6443	-0.1128

C11

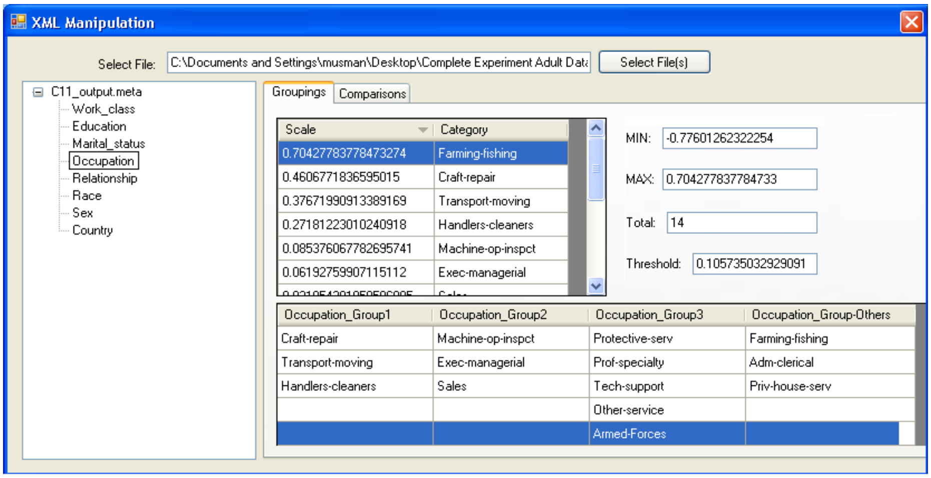
Rank	Variables	Entropy	Info.Gain
1	Occupation	2.5032	0.2458
2	Marital_Status	1.7637	0.1837
3	Relationship	1.9823	0.0770
4	Work_Class	1.1009	0.0215
5	Sex	0.9182	0.0121
6	Race	0.8042	0.0063
7	Education	2.3019	-0.0177
8	Country	0.5887	-0.1362

C12

Rank	Variables	Entropy	Info. Gain
1	Education	2.2195	0.8245
2	Work_Class	1.6272	0.5340
3	Race	0.5717	0.1202
4	Occupation	1.5472	0.1193
5	Sex	0.5714	0.0429
6	Marital_Status	0.6705	0.0174
7	Country	0.3771	0.0065
8	Relationship	0.7891	-0.0655

Fig.14. Ranked lists of nominal variables present in Adult dataset clusters

In the next step, we applied multidimensional scaling and group values present in each nominal variable using our developed prototype. Figure 15 shows the results of the nominal variable *Occupation* present in cluster C11 in the interface of our developed prototype.



Select File: C:\Documents and Settings\musman\Desktop\Complete Experiment Adult Data

Scale	Category
0.70427783778473274	Farming-fishing
0.4606771836595015	Craft-repair
0.37671990913389169	Transport-moving
0.27181223010240918	Handlers-cleaners
0.085376067782695741	Machine-op-inspct
0.06192759907115112	Exec-managerial

MIN: -0.77601262322254

MAX: 0.704277837784733

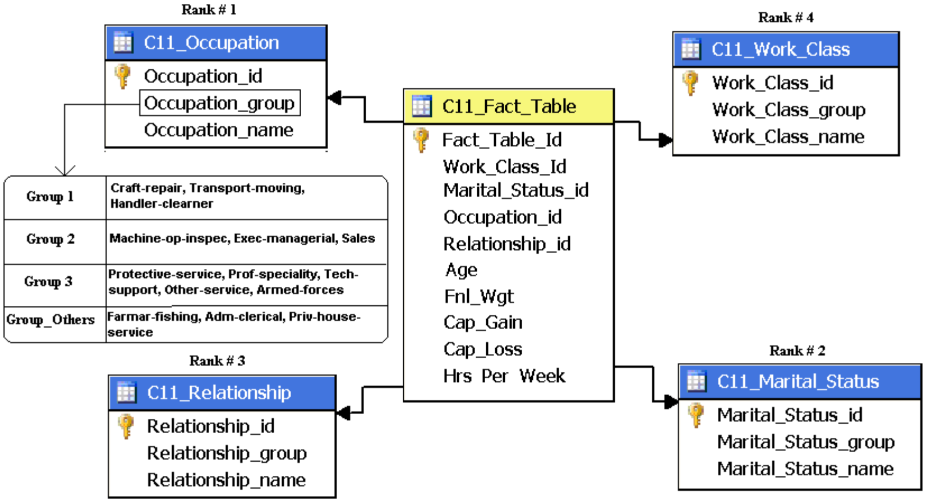
Total: 14

Threshold: 0.105735032929091

Occupation_Group1	Occupation_Group2	Occupation_Group3	Occupation_Group-Others
Craft-repair	Machine-op-inspct	Protective-serv	Farming-fishing
Transport-moving	Exec-managerial	Prof-specialty	Adm-clerical
Handlers-cleaners	Sales	Tech-support	Priv-house-serv
		Other-service	
		Armed-Forces	

Fig.15. Groupings achieved for cluster C11 through our developed prototype for cluster C11

The generated schema shown in Figure 16 contains 4 highly ranked dimensions, namely *Occupation*, *Marital_Status*, *Relationship* and *Work_Class*. Moreover, the natural groups obtained in the previous step of our methodology are also depicted for the top ranked *Occupation* dimension.



Group 1	Craft-repair, Transport-moving, Handler-clearner
Group 2	Machine-op-inspec, Exec-managerial, Sales
Group 3	Protective-service, Prof-speciality, Tech-support, Other-service, Armed-forces
Group_Others	Farmar-fishing, Adm-clerical, Priv-house-service

Fig.16. Multidimensional schema generated for cluster C11

The natural groupings highlight the semantic relationships present in various occupations. For instance, people having occupations like (*craft-repair*, *transport-moving* and *handler cleaner*) tend to have similar behaviour and are grouped together in cluster C11. It would be interesting to examine the distribution of fact variables such as *Hrs-per-week* or *Age* associated with these occupation types.

In order to explore these rich dimensions and facts, we construct a 3-dimensional cube using the 3 top ranked dimensions and facts. Each dimension has two levels (group level and value level) in order to navigate in the dimensional hierarchy. Figure 17 shows the 3 dimensional cubes structure of two clusters at different levels of the hierarchy.

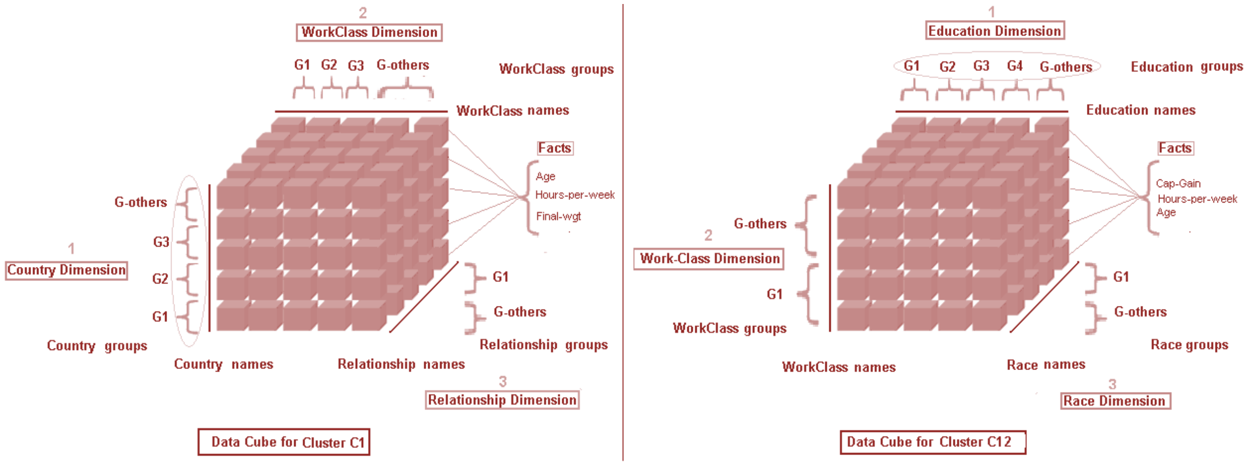


Fig.17. Differences in dimension rankings and groupings in cluster C1 and C12

It can be seen from Figure 17 that *Relationship* happens to be the 3rd ranked dimension in cluster C1; however, is totally absent in a lower level cluster C12. Moreover, the *Work-Class* dimension remained at number 2 in rank on both cubes. However, the number of groups within this dimension changed significantly. The *WorkClass* dimension has 4 groups [G1, G2, G3 and G-others] in the cluster C1 cube whereas the same dimension has only 2 groups [G1 and G-others] in the cluster C12 cube which is at a lower level in the cluster hierarchy. Additionally, the values present in each group appear to be entirely different from one another. Such sharp differences give new insights about the underlying data and further relations of these dimensions with the facts can be explored in an OLAP manner.

As done in the previous case study, we applied rule mining on both original cluster data and multidimensional schema for clusters C1, C11 and C12. Similar to the previous case study, we picked the three most impure dimensions (having high entropy values) to generate association rules. Table 11 shows the top 10 rules of cluster C12 generated without multidimensional schema at a minimum probability value of 0.4 and minimum importance value of 0.10. Based on these thresholds the algorithm produced 53 rules in total.

Table 11. Rules generated without multidimensional schema

RULES WITHOUT SCHEMA (CLUSTER C12)		
No	**Rules**	**Importance**
R1	Education = Bachelors, Occupation = Adm-clerical → Work Class = Local-gov	1.343
R2	Education = [*All*] ,Occupation = Handlers-cleaners → Work Class = Local-gov	1.298
R3	Education = 10th, Occupation = [*All*] → Work Class = Local-gov	1.170
R4	Education = HS-grad, Occupation = Adm-clerical → Work Class = Self-emp-not-inc	0.535
R5	Education = Some-college, Occupation = Sales → Work Class = Self-emp-not-inc	0.487
R6	Education = HS-grad, Occupation = Prof-specialty → Work Class = Self-emp-not-inc	0.407
R7	Education = Bachelors, Occupation = Sales →Work Class = Self-emp-inc	0.368
R8	Education = HS-grad, Occupation = Exec-managerial →Work Class = Self-emp-not-inc	0.346
R9	Education = HS-grad, Occupation = Sales →Work Class = Self-emp-inc	0.275
R10	Education = Some-college, Occupation = Exec-managerial →Work Class = Private	0.190

With the same threshold values, we generated the rules from the multidimensional schema of the same cluster, C12. The first 10 rules satisfied the given thresholds and are shown in Table 12.

Table 12. Rules generated with multidimensional schema

RULES WITH SCHEMA (CLUSTER C12)		
No	**Rules**	**Importance**
R1	Education Group = group1, Occupation Group = group1 → Work Class Name = Local-gov	1.343
R2	Education Group = [*All*], Occupation Group = group_others → Work Class Name = Local-gov	1.298
R3	Education Group = group_others , Occupation Group = [*All*] →Work Class Name = Local-gov	1.170
R4	Education Group = group_others , Occupation Group = group1→Work Class Name = Self-emp-not-inc	0.535
R5	Education Group = group1, Occupation Group = group1 → Work Class Name = Self-emp-not-inc	0.487
R6	Education Group = group1, Occupation Group = group1→ Work Class Name = Self-emp-inc	0.368
R7	Education Group = group_others, Occupation Group = group1→ Work Class Name = Self-emp-inc	0.275
R8	Education Group = group1, Occupation Group = group1→ Work Class Name = Private	0.190
R9	Education Group = [*All*], Occupation Group = group1→ Work Class Name = Private	0.159
R10	Education Group = group_others , Occupation Group = group_others→ Work Class Name = Private	0.154

We observe that the rules generated with the use of the multidimensional schema follow the same trends as in the previous case study. The rules generated through the use of the multidimensional schema are more informative. For example, Rule (R1) without schema predicts Work-Class = [Local-gov] if Education = [Bachelors] and Occupation = [Adm-clerical]. On the other hand, Rule (R1) with schema predicts the same Work-Class value with a diverse set of Education and Occupation values present in [group1] of each dimension. For instance, [group1] of Education consists of (Doctorate, Masters, Bachelors, Prof-school, Some-college, Assoc-voc) and [group1] of Occupation consists of (Prof-specialty, Exec-managerial, Tech-support, Sales, Adm-clerical, Protective-serv, Other-service, Transport-moving, Machine-op-inspct). These groups provide rich and diverse information to the user while retaining the same importance score of 1.343. If we focus on the Education dimension, we see that rule R1 without schema only suggests Bachelor as the education for people who work in Local-government, whereas R1 with the schema suggests a set of values in which Masters and Doctorate qualifications are also present.

In order to validate the claim that rules generated from schema are more diverse, we performed evaluation of the rules with the *Rae*, *CON* and *Hill* diversity measures, as with the previous Case Study. We took a set of top rules generated without schema and compared it with the same set of rules generated with the use of the schema. Table 13 shows the results of this evaluation.

Table 13. Rule interestingness comparison using objective diversity measures

Cluster labels	Rule set	NO Schema Rae	With Schema Rae	NO Schema CON	With Schema CON	NO Schema Hill	With Schema Hill
C1	**R1-R6**	0.322	**0.337**	0.432	**0.452**	-1.693	**-1.636**
	R1-R7	0.259	**0.295**	0.369	**0.421**	-2.302	**-1.948**
	R1-R8	0.218	**0.277**	0.327	**0.418**	-2.873	**-2.095**
	R1-R9	0.204	**0.270**	0.325	**0.423**	-3.084	**-2.159**
	R1-R10	0.192	**0.264**	0.321	**0.427**	-3.285	**-2.216**
C11	**R1-R6**	0.322	**0.397**	0.432	**0.526**	-1.693	**-1.286**
	R1-R7	0.256	**0.376**	0.365	**0.522**	-2.346	**-1.385**
	R1-R8	0.216	**0.312**	0.324	**0.463**	-2.918	**-1.805**
	R1-R9	0.204	**0.309**	0.325	**0.472**	-3.084	**-1.829**
	R1-R10	0.192	**0.303**	0.321	**0.475**	-3.285	**-1.872**
C12	**R1-R6**	0.164	**0.263**	0.263	**0.371**	-3.017	**-2.248**
	R1-R7	0.217	**0.312**	0.353	**0.454**	-2.437	**-1.703**
	R1-R8	0.172	**0.297**	0.294	**0.454**	-3.339	**-1.800**
	R1-R9	0.142	**0.279**	0.252	**0.441**	-4.261	**-2.110**
	R1-R10	0.136	**0.276**	0.448	**0.443**	-4.755	**-2.141**

It is apparent from Table 13 that all the objective measures of diversity show significant improvement for the set of rules generated from multidimensional schema. Similar to the *Automobile* case study results, the rules generated from the multidimensional schema are more diverse and capable of conveying more interesting knowledge to the user. Furthermore, the prediction accuracy of the rules generated from the schema also appears to be higher when compared to the rules without schema. To validate this claim, we tested the prediction accuracy of the rules generated without schema against the rules generated with schema. We used 30% of the test data from each cluster and ran 10 tests where each test randomly picked unique test data to assess the predictive accuracy of association rules generated. Table 14 shows the prediction accuracy for each test, expressed as a percentage.

Table 14. Rule prediction accuracies for the two sets of rules

Prediction Tests	Cluster C1		Cluster C11		Cluster C12	
	Schema	**No Schema**	**Schema**	**No Schema**	**Schema**	**No Schema**
Test 1	39.7	40.1	40.1	40.1	50	52.2
Test 2	40	40.3	40.3	40	47.7	47.7
Test 3	40.2	39.7	39.7	39.1	56.8	40.9
Test 4	39.5	40	40	39.3	47.7	38.6
Test 5	41.5	39.5	39.5	40.8	43.1	50
Test 6	40	39.7	40.3	39	55.3	40.9
Test 7	41.5	40.1	39.7	39.7	56.8	50.2
Test 8	39.7	39.5	40.1	40	50	39.3
Test 9	39.5	40	40	39.3	47.3	40.9
Test 10	40.2	39.5	41.3	40.8	47.7	47.7
Average Percentage	**40.18**	39.84	**40.01**	39.81	**50.24**	44.84

It is clear from Table 14 that the prediction accuracy of the rules generated through the use of the multidimensional schema is higher when compared to the one without schema. Again, we note from this case study on a relatively large dataset, that rule mining performed on the multidimensional schema designed and constructed with the help of hierarchical clustering and multidimensional scaling technique generates diverse rules with greater prediction accuracy.

5.3. Case study 3 – CoverType dataset

In this section, we present our third case study conducted on a much larger dataset called CoverType (Blackard et al., 1998). This is currently one of the largest datasets in the UCI repository containing 581012 records with 54 variables (42 nominal and 12 numeric) and 7 target classes (Obradovic and Vucetic, 2004). This benchmark dataset is used to predict forest cover types from cartographic variables. Forest cover type is basically defined as a descriptive classification of forest land based on occupancy of an area by the tree species present in it. It is a typical real world dataset having an imbalanced class distribution for the cover type variable.

The actual forest cover type for a given observation (30 x 30 meter cell) was determined from US Forest Service (USFS) Region 2 Resource Information System (RIS) database. Independent variables were derived from data originally obtained from the US Geological Survey (USGS) and USFS data. Data is in raw form (not scaled) and contains binary (0 or 1) columns of data for qualitative independent variables (wilderness areas and soil types). This study area includes four wilderness areas located in the Roosevelt National Forest of Northern Colorado. More detailed description and background information for this dataset can be found at University of California – machine learning website (Asuncion and Newman, 2010).

The main motivation behind choosing this dataset is that it poses extreme challenges to the analysts in finding useful and interesting knowledge from the rich mix of nominal and numeric variables and sheer size of data. To give a glimpse of the difficulties involved in mining association rules from this complex dataset, we present some interesting facts for this dataset discovered by (Webb, 2006) which motivated us to mine diverse association rules from this large dataset. (Webb 2006) reported the results of this particular dataset and observed that not a single non-redundant rule generated through *CoverType* dataset was found to be productive. He explained that this was due to a peculiarity of this dataset which uses 40 mutually exclusive binary variables (SoilType 1 to SoilType 40) to represent which one of 40 soil types predominates in an area. Thus, the most frequent attribute values are values of 0 for individual soil type variables and the most frequent itemsets are sets of these values. Because they are mutually exclusive, for any two of these variables, all associations between these variables must be unproductive. The identified fact that all non-redundant associations for this dataset represented unproductive associations highlighted the dangers of data mining without sound methodologies for discovering meaningful and diverse association rules.

Unlike with the two previous case studies this dataset was sampled prior to the application of hierarchical clustering. Stratified random sampling was used to obtain an unbiased sample in view of the unbalanced nature of the dataset. In this process we divided the records into homogeneous subgroups, defined by the forest cover type variable prior to sampling, thus improving the representativeness of each class in the sample. We used a sample size of 45,000 records to generate clusters at multiple levels of data abstraction. Similar to the previous case studies on smaller datasets, we calculated the inconsistency coefficient value which was 0.519 for determining the cut-off point and generated the binary tree of clusters using the procedure explained in Algorithm 1. There were a total of 8 levels of data abstraction till the cut-off point and the last level had 16 clusters in it.

Although we adopted a stratified sampling method to produce a sample that retains the diversity of classes, the sample size was rather small at approximately 8 % of the total volume of records. As a consequence, overall information loss could be high enough to prevent the accurate depiction of all trends and patterns which exist in the original dataset. In order to accelerate the creation of the dendrogram, we used the sampled dataset, instead of the original dataset. Once the binary tree was created, we distributed the original (non-sampled) data by allocating each record to the cluster whose centroid was closest the current record in Euclidean terms. The population of the dendogram with the entire dataset was done in order to alleviate the problem of small volume preventing the identification of certain patterns. Figure 18 depicts the first four levels of binary tree after allocation of records for the *CoverType* dataset.

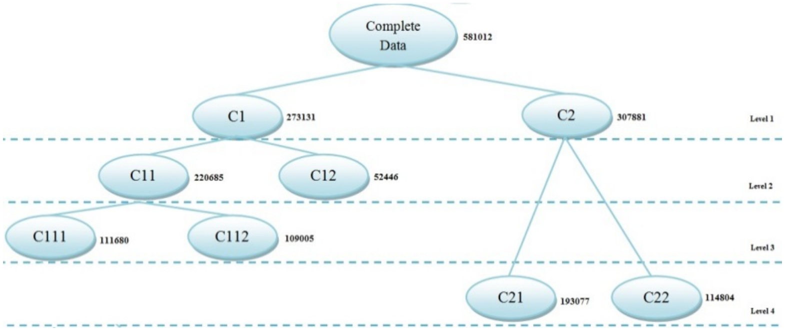
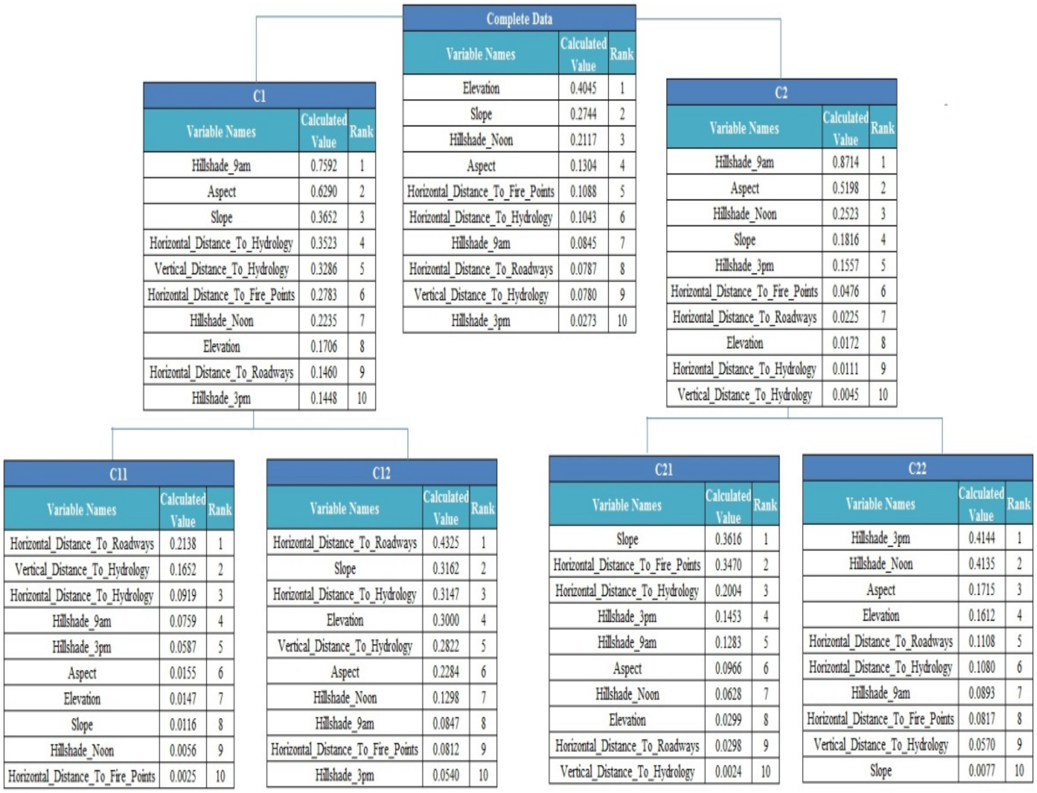


Fig.18. Binary tree for CoverType dataset

Likewise, as in the previous case studies, we plotted the numeric variables present in each cluster to fix the number of principal components to be extracted. Most clusters were discriminating well on 1 component. The application procedure remains the same as in the previous two case studies.

We rank the numeric variables present in each cluster and observe that the ranking is unique to each cluster. Similar to the previous case studies, sharp differences in rankings for a given variable appear at different levels in the cluster hierarchy. The rankings computed for the first few levels of the hierarchy are shown in Figure 19.



Complete Data

Variable Names	Calculated Value	Rank
Elevation	0.4045	1
Slope	0.2744	2
Hillshade_Noon	0.2117	3
Aspect	0.1304	4
Horizontal_Distance_To_Fire_Points	0.1088	5
Horizontal_Distance_To_Hydrology	0.1043	6
Hillshade_9am	0.0845	7
Horizontal_Distance_To_Roadways	0.0787	8
Vertical_Distance_To_Hydrology	0.0780	9
Hillshade_3pm	0.0273	10

C1

Variable Names	Calculated Value	Rank
Hillshade_9am	0.7592	1
Aspect	0.6290	2
Slope	0.3652	3
Horizontal_Distance_To_Hydrology	0.3523	4
Vertical_Distance_To_Hydrology	0.3286	5
Horizontal_Distance_To_Fire_Points	0.2783	6
Hillshade_Noon	0.2235	7
Elevation	0.1706	8
Horizontal_Distance_To_Roadways	0.1460	9
Hillshade_3pm	0.1448	10

C2

Variable Names	Calculated Value	Rank
Hillshade_9am	0.8714	1
Aspect	0.5198	2
Hillshade_Noon	0.2523	3
Slope	0.1816	4
Hillshade_3pm	0.1557	5
Horizontal_Distance_To_Fire_Points	0.0476	6
Horizontal_Distance_To_Roadways	0.0225	7
Elevation	0.0172	8
Horizontal_Distance_To_Hydrology	0.0111	9
Vertical_Distance_To_Hydrology	0.0045	10

C11

Variable Names	Calculated Value	Rank
Horizontal_Distance_To_Roadways	0.2138	1
Vertical_Distance_To_Hydrology	0.1652	2
Horizontal_Distance_To_Hydrology	0.0919	3
Hillshade_9am	0.0759	4
Hillshade_3pm	0.0587	5
Aspect	0.0155	6
Elevation	0.0147	7
Slope	0.0116	8
Hillshade_Noon	0.0056	9
Horizontal_Distance_To_Fire_Points	0.0025	10

C12

Variable Names	Calculated Value	Rank
Horizontal_Distance_To_Roadways	0.4325	1
Slope	0.3162	2
Horizontal_Distance_To_Hydrology	0.3147	3
Elevation	0.3000	4
Vertical_Distance_To_Hydrology	0.2822	5
Aspect	0.2284	6
Hillshade_Noon	0.1298	7
Hillshade_9am	0.0847	8
Horizontal_Distance_To_Fire_Points	0.0812	9
Hillshade_3pm	0.0540	10

C21

Variable Names	Calculated Value	Rank
Slope	0.3616	1
Horizontal_Distance_To_Fire_Points	0.3470	2
Horizontal_Distance_To_Hydrology	0.2004	3
Hillshade_3pm	0.1453	4
Hillshade_9am	0.1283	5
Aspect	0.0966	6
Hillshade_Noon	0.0628	7
Elevation	0.0299	8
Horizontal_Distance_To_Roadways	0.0298	9
Vertical_Distance_To_Hydrology	0.0024	10

C22

Variable Names	Calculated Value	Rank
Hillshade_3pm	0.4144	1
Hillshade_Noon	0.4135	2
Aspect	0.1715	3
Elevation	0.1612	4
Horizontal_Distance_To_Roadways	0.1108	5
Horizontal_Distance_To_Hydrology	0.1080	6
Hillshade_9am	0.0893	7
Horizontal_Distance_To_Fire_Points	0.0817	8
Vertical_Distance_To_Hydrology	0.0570	9
Slope	0.0077	10

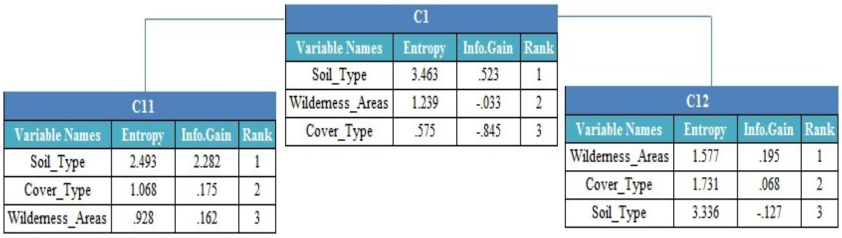
Fig.19. Ranked lists of numeric variables present in CoverType dataset clusters

Figure 19 shows the ranking of numeric variables in the cluster hierarchy after performing PCA and comparative analysis as explained in section 4.2. Unlike in the previous two case studies, we observe from Figure 19 that the clusters at the first level of the hierarchy, namely C1 and C2 have the same top 2 highly ranked variables. This is due to the component loadings of *Hillshade_9am* and *Aspect* variables in the four clusters, namely (C11, C12, C21 and C22), responsible for the ranking the two clusters, C1 and C2, at the first level of the hierarchy. These four clusters showed maximum difference in component loading values for *Hillshade_9am* and *Aspect* variables as compared to all other variables involved in the analysis. For instance, the component loadings of *Hillshade_9am* in C11 and C12 are 0.193 and 0.952 respectively which makes the difference equal to 0.759, which is the highest as compared to all the other variables and consequently ranks *Hillshade_9am* at the top in cluster C1.

Similarly the *Aspect* variable is ranked 2nd in cluster C2 because the two subsequent clusters, namely C21 and C22, responsible for determining this rank have loading values of 0.605 and 0.085, correspondingly making the difference in loadings equals to 0.519 which is the 2nd highest difference after the highest difference of 0.871 for the *Hillshade_9am* variable. Interestingly, this case study shows that in certain datasets such as CoverType, clear delineations between clusters only start to occur at levels 2 and below in the dendrogram structure. Despite this, the level 1 delineation is important as the subtle variations in level 1 are a necessary pre-requisite for uncovering more pronounced deviations further down in the cluster hierarchy.

Interestingly, the most significant variable *Elevation* in the complete un-clustered dataset is amongst the least significant variables at lower levels in the cluster hierarchy. This implies that forest cover types in the complete dataset are split into two separate clusters, primarily on the basis of the *Elevation* and *Slope* variables and marginally on the basis of other variables such as *Vertical_Distance_To_Hydrology* and *Hillshade_3pm.* Similarly, *Horizontal_Distance_To_Roadways* variable which happens to be an insignificant variable in cluster C1 appears to be the most significant in the child clusters C11 and C12. Another strong pattern can be observed in the rankings of cluster C21 and C22, whereby the *Slope* variable is ranked first in terms of significance in one child cluster C21 and whilst being the least significant in the other child cluster C22 at the same hierarchical level.

After establishing the ranking of numeric variables, we rank the nominal variables in each cluster based on the information gain measure. Figure 20 shows the information gain based ranking for three clusters, at consecutive levels in the hierarchy, namely clusters C1, C11 and C12.



C1

Variable Names	Entropy	Info.Gain	Rank
Soil_Type	3.463	.523	1
Wilderness_Areas	1.239	-.033	2
Cover_Type	.575	-.845	3

C11

Variable Names	Entropy	Info.Gain	Rank
Soil_Type	2.493	2.282	1
Cover_Type	1.068	.175	2
Wilderness_Areas	.928	.162	3

C12

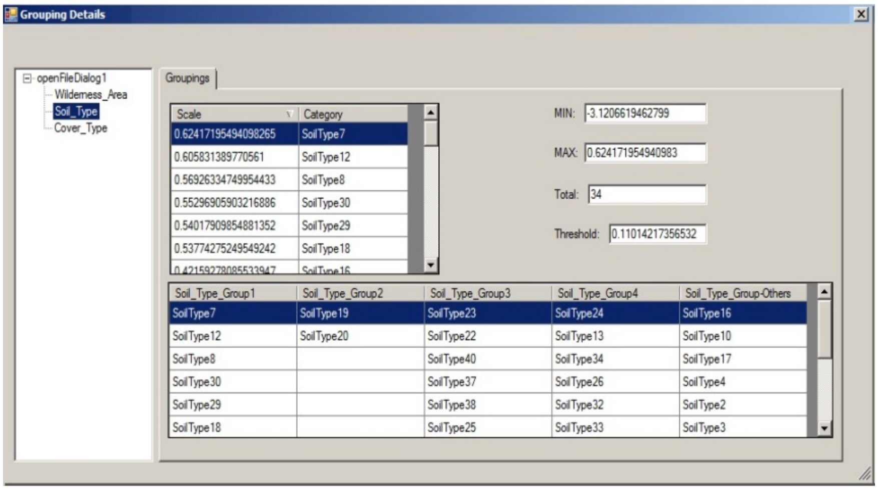
Variable Names	Entropy	Info.Gain	Rank
Wilderness_Areas	1.577	.195	1
Cover_Type	1.731	.068	2
Soil_Type	3.336	-.127	3

Fig.20. Ranking of nominal variables based on Information Gain measure

Similar to numeric variables, the list of ranked nominal variables also show sharp differences as we move from a higher level of data abstraction to a lower level. It can be seen from Figure 20 that *Soil_Type* which is top-ranked in cluster C1 is lowest ranked in cluster C12. Similarly, *Wilderness_Area* which is the second most significant variable in C1 took last place in cluster C11. In other words, *Soil_Type* and *Wilderness_Area* had high randomness in parent cluster C1 but at an immediately lower level in the hierarchy these variables appear to be having low randomness or impurity.

Likewise previous case studies, we observe the emergence of sharp patterns in both numeric and nominal variables as we go down in the cluster hierarchy. We suggest that these highly ranked variables should be explored as a first choice in a given cluster C1 as they possess more versatile information that defines the split as compared to the lowly ranked variables which play a minimum role in the cluster split.

After ranking the nominal variables, we apply multidimensional scaling and group the values present in each nominal variable using Algorithm 2. Figure 21 shows the groupings obtained for the *Soil_Type* variable in C11.



Scale	Category
0.62417195494098265	SoilType7
0.605831389770561	SoilType12
0.56926334749954433	SoilType8
0.55296905903216886	SoilType30
0.54017909854881352	SoilType29
0.53774275249549242	SoilType18
0.42159278085533947	SoilType16

MIN: -3.1206619462799

MAX: 0.624171954940983

Total: 34

Threshold: 0.11014217356532

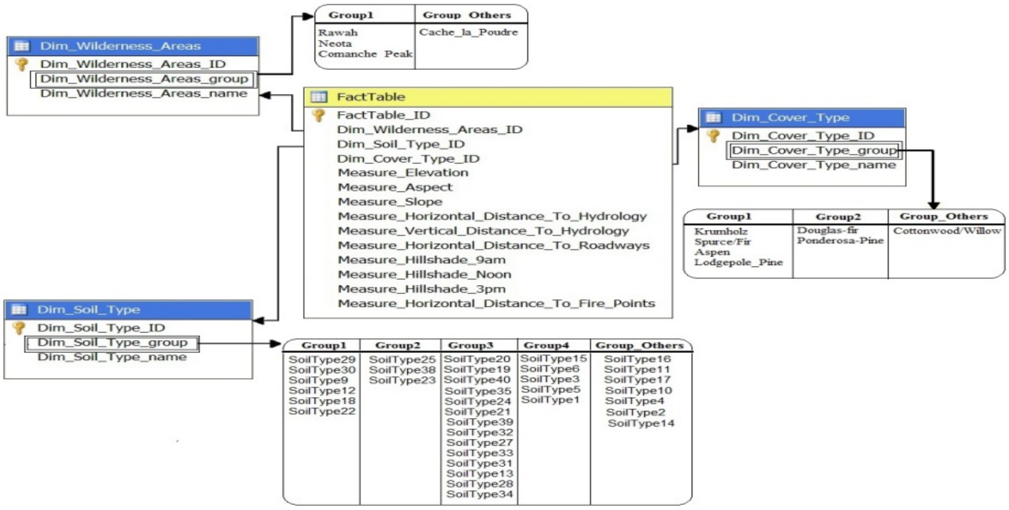
Soil_Type_Group1	Soil_Type_Group2	Soil_Type_Group3	Soil_Type_Group4	Soil_Type_Group-Others
SoilType7	SoilType19	SoilType23	SoilType24	SoilType16
SoilType12	SoilType20	SoilType22	SoilType13	SoilType10
SoilType8		SoilType40	SoilType34	SoilType17
SoilType30		SoilType37	SoilType26	SoilType4
SoilType29		SoilType38	SoilType32	SoilType2
SoilType18		SoilType25	SoilType33	SoilType3

Fig.21. Groupings achieved for Soil_Type variable in cluster C11

The prototype displays the different groups and the values present for each group in a data grid for a selected variable. It can be seen from Figure 21 that the *Soil_Type* variable produced five distinct major groups covering the 34 different soil types present in the cluster C11.

Furthermore, each group represents soil type values which have semantic relationships with other values present in the same group. For instance, Group 2 consists of soil type 19 and soil type 20 which are grouped together and both of these soil types have the same metadata description in the dataset. These soil types belongs to "Typic Cryaguolis and Typic Cryaquolls", two families of soil types which have the same depth (0 to 4 inches) and same dominant plant (subalpine fir) associations. Moreover, their climate and geographical zones are also identical. It highlights the fact that groups created through the use of multidimensional scaling technique not only have objective similarities but also have likeness in a real-world setting.

After applying multidimensional scaling technique for grouping similar values, we generate a multi-dimensional schema by using the numerical variables as facts and nominal variables as dimensions. The ranked lists of numeric and nominal variables serve as a starting point for selection of an appropriate number of important dimensions (nominal variables) and facts (numeric variables) for the creation of the multidimensional schema. Figure 22 shows the schema generated for cluster C22 using the three dimensions, namely, *Wilderness Area*, *Soil_Type* and *Cover_Type*.



Dim_Wilderness_Areas
Dim_Wilderness_Areas_ID
Dim_Wilderness_Areas_group
Dim_Wilderness_Areas_name

Group1	Group Others
Rawah Neota Comanche Peak	Cache_la_Poudre

FactTable
FactTable_ID
Dim_Wilderness_Areas_ID
Dim_Soil_Type_ID
Dim_Cover_Type_ID
Measure_Elevation
Measure_Aspect
Measure_Slope
Measure_Horizontal_Distance_To_Hydrology
Measure_Vertical_Distance_To_Hydrology
Measure_Horizontal_Distance_To_Roadways
Measure_Hillshade_9am
Measure_Hillshade_Noon
Measure_Hillshade_3pm
Measure_Horizontal_Distance_To_Fire_Points

Dim_Cover_Type
Dim_Cover_Type_ID
Dim_Cover_Type_group
Dim_Cover_Type_name

Group1	Group2	Group_Others
Krumholz Spurce/Fir Aspen Lodgepole_Pine	Douglas-fir Ponderosa-Pine	Cottonwood/Willow

Dim_Soil_Type
Dim_Soil_Type_ID
Dim_Soil_Type_group
Dim_Soil_Type_name

Group1	Group2	Group3	Group4	Group_Others
SoilType29 SoilType30 SoilType9 SoilType12 SoilType18 SoilType22	SoilType25 SoilType38 SoilType23	SoilType20 SoilType19 SoilType40 SoilType35 SoilType24 SoilType21 SoilType39 SoilType32 SoilType27 SoilType33 SoilType31 SoilType13 SoilType28 SoilType34	SoilType15 SoilType6 SoilType3 SoilType5 SoilType1	SoilType16 SoilType11 SoilType17 SoilType10 SoilType4 SoilType2 SoilType14

Fig.22. Multidimensional schema generated for cluster C22

Moreover, the natural groups obtained through multidimensional scaling are also depicted for each dimension. The natural groupings highlight the semantic relationships present in various wilderness areas, cover types and soil types. For instance, forest land having soil types *(SoilType23, SoilType25 and SoilType38)* tend to have similar behaviour and are therefore grouped together in Group2 of cluster C22.

In the detailed description of the soil types, present on the UCI machine learning website (Asuncion and Newman, 2010), it is evident that the above three soil types belong to the *Leighcan family* of soils. According to United States Department of Agriculture (USDA, 2012), soils from the *Leighcan family* occur on moraines and consist of residuum and/or till from igneous and metamorphic rock. These soils share common characteristics such as extremely warm climate and stony geographical zones. Additionally, these particular soil types lie in the same climatic zone of *Subalpine.* In the National Cooperative Soil Survey conducted by US National Resource Conservation Services (NRCS, 2012), these zones have (0 to 15%) of slopes, (7000 to 12,000) feet elevation and are derived from gravel deposits of igneous, metamorphic and sedimentary rocks. These similarities among the soil types highlight the fact that groups created through the use of multidimensional scaling technique not only have objective similarities but also have correspondence in a real-life. In this context it would be productive to explore and analyze the distributions of fact variables such as *Elevation* or *Slope* associated with these semantically related groups of soil types, wilderness areas and cover types.

In order to explore these ranked dimensions and facts, we construct a 3-dimensional cube using the dimensions and top 3 facts. Figure 23 shows the 3 dimensional cube structure of two clusters at different levels of the hierarchy.

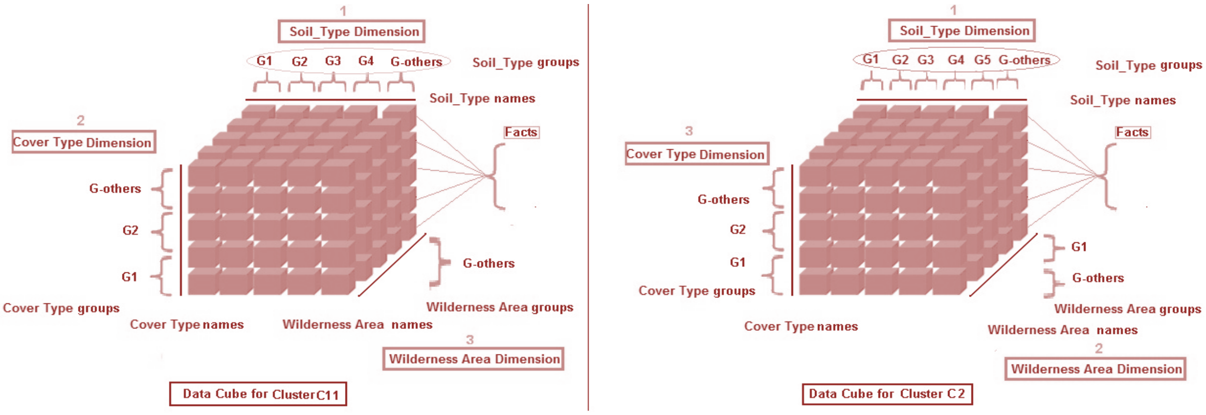


Fig.23. Comparison of 3-dimensional cubes at different levels of hierarchy

It can be seen from Figure 23 that *Wilderness Area* happens to be the 3rd ranked dimension in cluster C11; however, it holds 2nd position in an upper level cluster C2. Moreover, the *Soil Type* dimension remained at number 1 in rank on both cubes. However, both the total number and composition of groups within this dimension changed between clusters. The *Soil Type* dimension has 5 groups [G1, G2, G3, G4 and G-others] in the cluster C11 cube whereas the same dimension has an extra group [G5] in the cluster C2 cube which is at an upper level in the cluster hierarchy. Similarly, *Wilderness Area* has 2 groups [G1 and G-others] in cluster C2 cube but has only one group of outliers [G-others] in cluster C11 data cube. Additionally, the values present in each group are entirely different from one another. For instance, Group1 of *CoverType* dimension has four values [*Krummholz, Spruce/Fir, Lodgepole Pine* and *Aspen*] in C2 whereas the comparable Group1 in C11 data cube contains only 2 values [*Ponderosa Pine* and *Douglas-fir*]. Such sharp differences give new insights about the underlying data and further relationships of these dimensions with the fact variables can be explored through OLAP analysis.

As done in the previous case studies, we applied rule mining on both the original dataset as well as the multidimensional schema version for clusters C1, C11 and C12. We chose the two most impure dimensions (having high entropy values) to generate association rules. Table 15 shows the top 10 rules for cluster C12 generated without multidimensional schema at a minimum probability value of 0.41 and minimum importance value of 0.10. Based on these thresholds the algorithm produced 84 rules in total.

Table 15. Rules generated without multidimensional schema

RULES WITHOUT SCHEMA (CLUSTER C12)		
No	**Rules**	**Importance**
R1	Soil Type = SoilType3, Wilderness Areas = Cache_la_Poudre → Cover Type = Cottonwood/Willow	1.865
R2	Soil Type = SoilType17, Wilderness Areas = Cache_la_Poudre → Cover Type = Cottonwood/Willow	1.863
R3	Soil Type = SoilType38, Wilderness Areas = Comanche_Peak → Cover Type = Krummholz	1.605
R4	Soil Type = SoilType39, Wilderness Areas = Comanche_Peak → Cover Type = Krummholz	1.573
R5	Soil Type = SoilType40, Wilderness Areas = Comanche_Peak → Cover Type = Krummholz	1.537
R6	Soil Type = SoilType38, Wilderness Areas = Neota → Cover Type = Krummholz	1.486
R7	Soil Type = SoilType40, Wilderness Areas = Neota → Cover Type = Krummholz	1.382
R8	Soil Type = SoilType14, Wilderness Areas = [All] → Cover Type = Douglas-fir	1.020
R9	Soil Type = SoilType2, Wilderness Areas = [All] → Cover Type = Ponderosa-Pine	0.832
R10	Soil Type = SoilType10, Wilderness Areas = Cache_la_Poudre → Cover Type = Ponderosa-Pine	0.823

With the same threshold values, we generated the rules from the multidimensional schema for the same cluster, C12. The first 10 rules satisfied the given thresholds and are shown in Table 16.

Table 16. Rules generated with multidimensional schema

RULES WITH SCHEMA (CLUSTER C12)		
No	**Rules**	**Importance**
R1	Soil Type Group = Group4, Wilderness Area = Group-Others → Cover Type = Cottonwood/Willow	1.865
R2	Soil Type Group = Group3, Wilderness Area = Group-Others → Cover Type = Cottonwood/Willow	1.863
R3	Soil Type Group = Group1, Wilderness Area = Group1→ Cover Type = Krummholz	1.605
R4	Soil Type Group = [All], Wilderness Area Group = Group-Others → Cover Type = Ponderosa-Pine	1.065
R5	Soil Type Group = Group-Others, Wilderness Area Group = [All] → Cover Type = Douglas-fir	1.020
R6	Soil Type Group = Group4, Wilderness Area Group = Group-Others → Cover Type = Ponderosa-Pine	0.823
R7	Soil Type Group = Group3, Wilderness Area Group = [All] → Cover Type = Ponderosa-Pine	0.814
R8	Soil Type Group = Group5, Wilderness Area Group = Group-Others → Cover Type = Ponderosa-Pine	0.808
R9	Soil Type Group = Group4, Wilderness Area Group = Group1→ Cover Type = Ponderosa-Pine	0.597
R10	Soil Type Group = Group2, Wilderness Area Group = Group1 → Cover Type = Spruce-fir	0.504

We observe that the rules generated with the use of the multidimensional schema follow the same trends as in the previous case studies. The rules generated through the use of the multidimensional schema are more informative**.** These groups provide rich and diverse information to the user while retaining the same importance score. If we focus on the Wilderness-Area dimension, we see that rule R3 without schema only identifies Comanche_Peak as a forest land with soil type 38, whereas R3 with the schema identifies a diverse set of wilderness areas in which Rawah and Neota areas are also present.

In order to validate the claim that rules generated from schema are more diverse, we performed evaluation of the rules with the Rae, CON and Hill diversity measures, as with the previous case studies. We took a set of top rules generated without schema and compared it with the same set of rules generated with the use of the schema. Table 17 shows the results of this evaluation.

Table 17. Rule interestingness comparison using objective diversity measures

Cluster labels	**Rule set**	**NO Schema Rae**	**With Schema Rae**	**NO Schema CON**	**With Schema CON**	**NO Schema Hill**	**With Schema Hill**
C1	**R1-R6**	0.230	**0.278**	0.277	**0.366**	-2.987	**-2.293**
	R1-R7	0.239	**0.256**	0.342	**0.364**	-2.646	**-2.510**
	R1-R8	0.226	**0.240**	0.34	**0.363**	-2.77	**-2.712**
	R1-R9	0.184	**0.238**	0.288	**0.378**	-3.764	**-2.731**
	R1-R10	0.176	**0.234**	0.292	**0.386**	-4.005	**-2.780**
C11	**R1-R6**	0.275	**0.296**	0.361	**0.394**	-2.338	**-1.884**
	R1-R7	0.247	**0.281**	0.351	**0.402**	-2.644	**-1.999**
	R1-R8	0.228	**0.244**	0.344	**0.369**	-2.892	**-2.393**
	R1-R9	0.187	**0.229**	0.292	**0.364**	-3.884	**-2.573**
	R1-R10	0.161	**0.196**	0.258	**0.327**	-4.81	**-3.131**
C12	**R1-R6**	0.219	**0.284**	0.251	**0.376**	-3.304	**-2.294**
	R1-R7	0.181	**0.278**	0.208	**0.397**	-4.273	**-2.353**
	R1-R8	0.163	**0.263**	0.209	**0.397**	-4.749	**-2.509**
	R1-R9	0.149	**0.225**	0.208	**0.359**	-5.195	**-3.015**
	R1-R10	0.137	**0.205**	0.203	**0.342**	-5.757	**-3.348**

It is apparent from Table 17 that all the objective measures of diversity show significant improvement for the set of rules generated from the multidimensional schema. Similar to the *Automobile* and *Adult* case study results, the rules generated from the multidimensional schema are more diverse and capable of conveying more interesting knowledge to the user.

Furthermore, the prediction accuracy of the rules generated from the schema also appears to be higher when compared to the rules without schema. To validate this claim, we tested the prediction accuracy of the rules generated without schema against the rules generated with the use of the schema. We used 30% of the test data from each cluster and ran 10 tests where each test randomly picked unique test data to assess the predictive accuracy of association rules generated. Table 18 shows the prediction accuracy for each test, expressed as a percentage.

Table 18. Rule prediction accuracies for the two sets of rules

Prediction Tests	**Cluster C1**		**Cluster C11**		**Cluster C12**	
	Schema	**No Schema**	**Schema**	**No Schema**	**Schema**	**No Schema**
Test 1	64.52	64.54	66.28	66.22	64.25	64.15
Test 2	64.47	57.36	66.46	51.75	64.43	53.66
Test 3	64.70	64.45	66.10	51.47	64.22	53.21
Test 4	64.41	57.85	66.41	66.25	64.34	53.65
Test 5	64.50	64.32	66.25	51.81	64.54	64.26
Test 6	64.62	57.37	66.54	51.82	64.43	53.73
Test 7	64.89	57.55	66.42	66.26	64.33	64.27
Test 8	64.70	64.48	66.82	66.31	64.25	64.23
Test 9	64.71	64.22	66.11	51.47	64.62	53.46
Test 10	64.52	64.38	66.41	51.67	64.22	64.14
Average Percentage	**64.60**	61.65	**66.38**	57.50	**64.36**	58.88

It is clear from Table 18 that the prediction accuracy of the rules generated through the use of the multidimensional schema is higher when compared to the one without schema. Again, we note from this case study on a very large dataset, that rule mining performed on the multidimensional schema designed and constructed with the help of hierarchical clustering and multidimensional scaling technique generates diverse rules with greater prediction accuracy.

The three case studies performed on real world datasets have completely different ratios of nominal and numeric attributes. For instance, *Automobile* dataset has a nominal to numeric ratio of 11 to 15, *Adult* dataset has 8 to 5 and *CoverTyp*e dataset has 42 to 12. However, all these different percentages of nominal and numeric attributes do not affect the improvement in diversity measures. The results provide concrete evidence that all the case studies performed on different datasets and on multiple clusters have shown consistently better results on the objective measures of diversity regardless of the ratio of numeric and nominal attributes. However, the degree of improvement in diversity or quality of the diverse rules generated through multidimensional schema depends on the particular dataset and is beyond the scope of this paper.

6. Conclusion and future work

In this paper, we presented a methodology for designing a multidimensional schema and discovery of interesting and useful knowledge in the form of association rules. A multidimensional schema was designed by filtering out non informative dimensions and facts from an initial set of candidate dimensions and fact variables. This filtration was based on the use of Entropy, and PCA. This process allowed us to mine association rules using only the most informative dimensions and facts. The rules generated with our proposed methodology were shown to be more diverse on the *Rae*, *CON* and *Hill* measures when compared against an approach that simply generated rules without the use of such a multidimensional schema.

Moreover, the prediction accuracy of the rules generated with the use of the schema was shown to be higher than that obtained without the use of the schema. Our case studies also revealed that reliance on the use of standard rule evaluation measures such as *importance* can sometimes be misleading as certain rules were shown to be more diverse than others which had exactly the same importance score.

Our methodology is a step forward towards data-driven discovery systems and provides a set of techniques from the machine learning and statistical domains that coherently work together to facilitate the ultimate goal of knowledge discovery from large datasets. The proposed methodology achieves the three main objectives of this research. Firstly, it provides automated and data-driven support for the design and construction of multidimensional schema. Secondly, facilitates the generation of informative cubes at different levels of data abstraction and identified the effects of data abstraction levels on information content. Thirdly, it allows the discovery of diverse and predictive association rules from multidimensional cube structure at various levels of data abstraction.

Besides the advantages of the proposed system, there is an obvious limitation in terms of interestingness evaluation of rules. For instance, a number of other interestingness measures have been proposed in the literature to measure the interestingness of patterns such as novelty, generality, conciseness, peculiarity etc. However, our proposed methodology permits the evaluation of interestingness only through the diversity criterion. The aforementioned interestingness measures are all useful in different scenarios and are sometimes correlated with, rather independent of one another. For example, conciseness often coincides with generality and peculiarity may coincide with novelty. Conversely, some of these measures may have absolute independence, depending on the context or application domain. For example, diversity may have no correlation with novelty. Knowledge which may appear to be diverse does necessarily mean that it is novel.

It is extremely challenging to find truly novel patterns/rules from data, thus explaining why novelty has received the least attention in the research community (Geng and Hamilton, 2006). Novelty requires the knowledge discovery systems to model everything that the users know explicitly in order to detect what is unknown or novel. In general, it is not feasible for users to specify all knowledge quickly and consistently in a system. Therefore, the proposed methodology utilized probability-based objective measures for evaluation since such measures do not involve modelling user expectations in advance which is an extremely difficult task in areas where domain knowledge is either limited or not available.

At the moment the dimensions that we design support two level hierarchies, with the first level consisting of groups and the second consisting of individual values within each group. A promising direction for future research would be to explore the use of deeper hierarchies, as it was shown in the case studies that the dimensional structure was responsible for capturing rich information. Another possible direction for future work would be to find an automatic method for detecting that certain fact variables could in fact be used to define dimensions. For instance, the *Age* variable is inherently numeric but can be used as a dimension variable by discretizing its value into distinct age ranges. We also intend to test the methodology with complex datasets from the biological, medical and engineering domains to further test the performance and scalability of the proposed methodology.

References

ABDELBAKI, W., YAHIA, S. B. & MESSAOUD, R. B. 2012. NAP-SC: A Neural Approach for Prediction over Sparse Cubes. *Advanced Data Mining and Applications.* S. Zhou, S. Zhang and G. Karypis, Springer Berlin Heidelberg. 7713: 340-352.

AGRAWAL, R., IMIELIŃSKI, T. & SWAMI, A., 1993. Mining association rules between sets of items in large databases. ACM SIGMOD Record.

ASUNCION, A. & NEWMAN, D. J., 2010. UCI machine learning repository [http://archive.ics.uci.edu/ml] Irvine, CA: University of California, School of Information and Computer Science.

BEN MESSAOUD, R., LOUDCHER RABASEDA, S., ROKIA, M. & BOUSSAÏD, O., 2007. OLEMAR: an On-Line Environment for Mining Association Rules in Multidimensional Data. *Advances in Data Warehousing and Mining*, IGI Global, 2, 1-35.

BLACKARD, J. A., DEAN, D. & ANDERSON, C. 1998. Covertype dataset. [Online]http://archive.ics.uci.edu/ml/datasets/Covertype

CHUNG, S. M. & MANGAMURI, M., 2005. Mining association rules from the star schema on a parallel NCR teradata database system. *International Conference on Information Technology: Coding and Computing (ITCC'05).* Nevada

CORDES, D., HAUGHTON, V., CAREW, J. D., ARFANAKIS, K. & MARAVILLA, K., 2002. Hierarchical clustering to measure connectivity in fMRI resting-state data. *Magnetic Resonance Imaging*, 20**,** 305-317.

COX, M. A. A. & COX, T. F., 2008. Multidimensional scaling. *Handbook of Data Visualization***,** 315-347.

DORI, D., FELDMAN, R. & STURM, A., 2008. From conceptual models to schemata: An object-process-based data warehouse construction method. *Information Systems,* 33**,** 567-593.

GENG, L. & HAMILTON, H. J., 2006. Interestingness measures for data mining: A survey. *ACM Computing Surveys (CSUR),* 38**,** 9.

GOIL, S. & CHOUDHARY, A., 2001. PARSIMONY: An infrastructure for parallel multidimensional analysis and data mining. *Journal of Parallel and Distributed Computing,* 61**,** 285-321.

HAHN, K., SAPIA, C. & BLASCHKA, M., Automatically generating OLAP schemata from conceptual graphical models. Proceedings of the 3rd ACM international workshop on Data warehousing and OLAP, 2000. ACM, 9-16.

JINWOOK, S. & SHNEIDERMAN, B., 2002. Interactively exploring hierarchical clustering results [gene identification]. *Computer,* 35**,** 80-86.

KAMBER, M., HAN, J. & CHIANG, J. Y., 1997. Metarule-guided mining of multi-dimensional association rules using data cubes. 207-210.

KAYA, M. & ALHAJJ, R., 2003. Integrating fuzziness with OLAP association rules mining. *Machine Learning and Data Mining in Pattern Recognition***,** 65-81.

KOHAVI, R. & BECKER, B., 1996. *Adult dataset* [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Adult.

LIU, Z. & GUO, M., 2001. A proposal of integrating data mining and on-line analytical processing in data warehouse. *International Conferences on Info-tech and Info-net,* 3, 146-151.

MANSMANN, S., 2009. Extending the OLAP Technology to Handle Non-Conventional and Complex Data. *Journal of Computing Science and Engineering,* 1,125-160.

MESSAOUD, R. B., RABASÉDA, S. L., BOUSSAID, O. & MISSAOUI, R., 2006. Enhanced mining of association rules from data cubes. *DOLAP '06 Proceedings of the 9th ACM international workshop on Data warehousing and OLAP*. ACM, 11-18.

NAHAR, J., IMAM, T., TICKLE, K. S. & CHEN, Y.-P. P. 2013. Association rule mining to detect factors which contribute to heart disease in males and females. *Expert Systems with Applications,* 40**,** 1086-1093.

NESTOROV, S. & JUKIC, N., 2003. Ad-hoc association-rule mining within the data warehouse. *In Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS 2003),* 232–242.

NG, E. K. K., FU, A. W. C. & WANG, K., 2002. Mining Association Rules from Stars. *In IEEE International Conference on Data Mining (ICDM),* 322–329.

NKAMBOU, R., FOURNIER-VIGER, P. & NGUIFO, E. M., 2011. Learning task models in ill-defined domain using an hybrid knowledge discovery framework. *Knowledge-Based Systems,* 24**,** 176-185.

NRCS. 2012. *Official Soil Series Description - LEIGHCN series, National Cooperative Soil Survery, USA* [Online].

OBRADOVIC, Z. & VUCETIC, S. 2004. Challenges in Scientific Data Mining: Heterogeneous, Biased, and Large Samples. Technical Report, Center for Information Science and Technology, Temple University, Chapter 1, pp.1-24

OHMORI, T., NARUSE, M. & HOSHI, M., 2007. A New Data Cube for Integrating Data Mining and OLAP. *Data Engineering Workshop, IEEE 23rd International Conference,* 896-903.

PALOPOLI, L., PONTIERI, L., TERRACINA, G. & URSINO, D., 2002. A novel three-level architecture for large data warehouses* 1. *Journal of Systems Architecture,* 47**,** 937-958.

PARDILLO, J. & MAZÓN, J. N., 2010. Designing OLAP schemata for data warehouses from conceptual models with MDA. *Decision Support Systems,* 3**,** 51-62.

PARDILLO, J., ZUBCOFF, J., MAZÓN, J. N. & TRUJILLO, J., 2008. Applying MDA to integrate mining techniques into data warehouses: a time series case study. *Mining Multiple Information Sources MMIS 08***,** 47.

PEARS, R., KOH, Y. S., DOBBIE, G. & YEAP, W. 2013. Weighted association rule mining via a graph based connectivity model. *Information Sciences,* 218**,** 61-84.

PERALTA, V., MAROTTA, A. & RUGGIA, R., 2003. Towards the Automation of Data Warehouse Design. 15th Conference on Advanced Information Systems Engineering, short paper proceedings (CAISE FORUM).

PHALADIGANON, P., KIM, S. B., CHEN, V. C. P. & JIANG, W. 2013. Principal component analysis-based control charts for multivariate nonnormal distributions. *Expert Systems with Applications,* 40**,** 3044-3054.

POOLE, J. & MELLOR, D., 2001. *Common Warehouse Metamodel: An Introduction to the Standard for Data Warehouse Integration*, John Wiley & Sons, Inc.

PSAILA, G. & LANZI, P. L., 2000. Hierarchy-based mining of association rules in data warehouses. ACM, 307-312.

ROSARIO, G. E., RUNDENSTEINER, E. A., BROWN, D. C., WARD, M. O. & HUANG, S. 2004. Mapping nominal values to numbers for effective visualization. *Information Visualization,* 3**,** 80-95.

SAPIA, C., HÖFLING, G., MÜLLER, M., HAUSDORF, C., STOYAN, H. & GRIMMER, U., 1999. On supporting the data warehouse design by data mining techniques. Citeseer, 63.

SCHLIMMER, J. C., 1985. *Automobile dataset* http://archive.ics.uci.edu/ml/datasets/Automobile [Online]. [Accessed 20 june 2012].

SEO, J., BAKAY, M., CHEN, Y. W., HILMER, S., SHNEIDERMAN, B. & HOFFMAN, E. P., 2004. Interactively optimizing signal-to-noise ratios in expression profiling: project-specific algorithm selection and detection p-value weighting in Affymetrix microarrays. *Bioinformatics,* 20**,** 2534-2544.

SEO, J., BAKAY, M., ZHAO, P., CHEN, Y. W., CLARKSON, P., SHNEIDERMAN, B. & HOFFMAN, E. P., 2003. Interactive color mosaic and dendrogram displays for signal/noise optimization in microarray data analysis. Proceedings of the International Conference on Multimedia and Expo. 461-464.

SONG, I. Y., KHARE, R., AN, Y., LEE, S., KIM, S. P., KIM, J. & MOON, Y. S., 2008. Samstar: An automatic tool for generating star schemas from an entity-relationship diagram. *Conceptual Modeling-ER 2008*. 522-523.

TJIOE, H. C. & TANIAR, D., 2005. Mining association rules in data warehouses. *International Journal of Data Warehousing and Mining,* 1**,** 28-62.

TRYFONA, N., BUSBORG, F. & BORCH CHRISTIANSEN, J. G., 1999. starER: A conceptual model for data warehouse design. *Proceedings of the 2nd ACM International Workshop on Data Warehousing and OLAP (DOLAP)*. ACM, 3-8.

TRYFOS, P., 1998. *Methods for business analysis and forecasting: text and cases,* Wiley.

UGUZ, H., 2011. A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. *Knowledge-Based Systems,* 24**,** 1024-1032.

USDA. 2012. *Keys to soil taxonomy* [Online]. US Dept. of Agriculture, Natural Resources Conservation Service.

USMAN, M. & ASGHAR, S., 2011. An Architecture for Integrated Online Analytical Mining. *Journal of Emerging Technologies in Web Intelligence,* 3**,** 74-99.

USMAN, M., ASGHAR, S. & FONG, S. A Conceptual Model for Combining Enhanced OLAP and Data Mining Systems. INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference on, 25-27 Aug. 2009 2009. 1958-1963.

USMAN, M., ASGHAR, S. & FONG, S., 2010. Data Mining and Automatic OLAP Schema Generation. *Proc. of Fifth International conference on Digital Information Management (ICDIM),* 35-43.

USMAN, M. & PEARS, R., 2010. Integration of Data Mining and Data Warehousing: A Practical Methodology. *International Journal of Advancements in Computing Technology,* 2**,** 31-46.

USMAN, M. & PEARS, R., 2011. Multi Level Mining of Warehouse Schema. *In Networked Digital Technologies.* Communications in Computer and Information Science, Springer Berlin Heidelberg, 395-408.

USMAN, M., PEARS, R. & FONG, A. C. M., in press. A Data Mining Approach to Knowledge Discovery from Multidimensional Cube Structures. *Knowledge-Based Systems,* 40**,** 36-49.

WEBB, G. I. Discovering significant rules. Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006. ACM, 434-443.

YOU, J., DILLON, T. & LIU, J., 2001. An integration of data mining and data warehousing for hierarchical multimedia information retrieval. *International Symposium on Intelligent Multimedia, Video and Speech Processing,* 373-376.

ZBIDI, N., FAIZ, S. & LIMAM, M., 2006. On mining summaries by objective measures of interestingness. *Machine learning,* 62**,** 175-198.

ZHEN, L. & MINYI, G., 2001. A proposal of integrating data mining and on-line analytical processing in data warehouse. *Proceedings of the International Conference on Info-Tech and Info-Net*. 146-151.

ZHU, H., 1998. On-line analytical mining of association rules. *Master’s thesis,* Simon Fraser University, Burnaby, British Columbia, Canada.

ZUBCOFF, J., PARDILLO, J. & TRUJILLO, J., 2007. Integrating clustering data mining into the multidimensional modeling of data warehouses with UML profiles. *Data Warehousing and Knowledge Discovery***,** 199-208.