A virtual reality-based approach for interactive and visual mining of association rules

Zohra Ben Said

l'université nantes angers le mans
PÔLE DE RECHERCHE ET D'ENSEIGNEMENT SUPÉRIEUR

UNIVERSITÉ DE NANTES

Thèse de Doctorat

Zohra Ben Said - Guefrech

Mémoire présenté en vue de l'obtention du
grade de Docteur de l'Université de Nantes
sous le label de l'Université de Nantes Angers Le Mans

Discipline : Informatique
Spécialité : Génie Logiciel
Laboratoire : Laboratoire d'informatique de Nantes-Atlantique (LINA)

Soutenue le 25 octobre 2012

École doctorale : 503 (STIM)
Thèse n° : ED 503-175

A virtual reality-based approach for interactive and visual mining of association rules

JURY

Rapporteurs :	**M. Gilles VENTURINI**, Professeur, Ecole Polytechnique de l'Universite de Tours **M. Mustapha LEBBAH**, Maître de conférences - HDR, Université Paris 13
Examinateurs :	**M. Colin DE LA HIGUERA**, Professeur, Ecole Polytechnique de l'Universite de Nantes **M^me^ Hanene HAZZAG**, Maître de conférences, Université Paris 13
Invités :	**M. Julien BLANCHARD**, Maître de conférences, Ecole Polytechnique de l'Universite de Nantes **M. Fabien PICAROUGNE**, Maître de conférences , Ecole Polytechnique de l'Universite de Nantes
Directeur de thèse :	**M. Fabrice GUILLET**, Professeur, Ecole Polytechnique de l'Universite de Nantes
Co-encadrant de thèse :	**M. Paul RICHARD**, Maître de conférences - HDR, Université d'Angers

l'université nantes angers le mans
PÔLE DE RECHERCHE ET D'ENSEIGNEMENT SUPÉRIEUR

UNIVERSITÉ DE NANTES

Thèse de Doctorat

Zohra Ben Said - Guefrech

Mémoire présenté en vue de l'obtention du
grade de Docteur de l'Université de Nantes
sous le label de l'Université de Nantes Angers Le Mans

Discipline : Informatique
Spécialité : Génie Logiciel
Laboratoire : Laboratoire d'informatique de Nantes-Atlantique (LINA)

Soutenue le 25 octobre 2012

École doctorale : 503 (STIM)
Thèse n° : ED 503-175

A virtual reality-based approach for interactive and visual mining of association rules

JURY

Rapporteurs :	**M. Gilles VENTURINI**, Professeur, Ecole Polytechnique de l'Universite de Tours **M. Mustapha LEBBAH**, Maître de conférences - HDR, Université Paris 13
Examinateurs :	**M. Colin DE LA HIGUERA**, Professeur, Ecole Polytechnique de l'Universite de Nantes **M^me^ Hanene HAZZAG**, Maître de conférences, Université Paris 13
Invités :	**M. Julien BLANCHARD**, Maître de conférences, Ecole Polytechnique de l'Universite de Nantes **M. Fabien PICAROUGNE**, Maître de conférences , Ecole Polytechnique de l'Universite de Nantes
Directeur de thèse :	**M. Fabrice GUILLET**, Professeur, Ecole Polytechnique de l'Universite de Nantes
Co-encadrant de thèse :	**M. Paul RICHARD**, Maître de conférences - HDR, Université d'Angers

Abstract

This thesis is at the intersection of two active research areas: Association Rule Mining and Virtual Reality.

The main limitations of the association rule extraction algorithms are that (i) they produce large amount of rules and (ii) many extracted rules have no interest to the user.

In practise, the amount of generated rule sets limits severely the ability of the user to explore these rule sets in a reasonable time. In the literature, several solutions have been proposed to address this problem such as, post-processing of association rules. Post-processing allows rule validation and extraction of useful knowledge. Whereas rules are automatically extract by combinatorial algorithms, rule post-processing is done by user. Visualisation can help the user deal with large amount of data by representing it in visual form to improve cognition for acquisition and the use of new knowledge. In order to find relevant knowledge in visual representations, the decision-maker needs to freely rummage through large amount of data. Therefore it is essential to integrate him/her in the data mining process through the use of efficient interactive techniques. In this context, the use of Virtual Reality techniques is very relevant: it allows the user to quickly view and select rules that seem interesting.

This work addresses two main issues: the representation of association rules to allow user quickly detection of the most interesting rules and interactive exploration of rules. The first requires an intuitive metaphor representation of association rules. The second requires an interactive exploration process allowing the user searching interesting rules.

The main contributions of this work can be summarised as follows:

1. **Classification for Visual Data Mining based on both 3D representations and interaction techniques**

   We present and discuss the concepts of visualisation and visual data mining. Then, we present 3D representation and interaction techniques in the context of data mining. Furthermore, we propose a new classification for Visual Data Mining, based on both 3D representations and interaction techniques. Such a classification may help the user choose a visual representation and an interaction technique for a given application. This study allows us to identify limitations of the knowledge visualisation approaches proposed in the literature.

2. **Metaphor for association rule representation**

   We propose a new visualisation metaphor for association rules. This new metaphor takes into account more accurately the attributes of the antecedent and the consequent, the contribution of each one to the rule, and their correlations. This metaphor is based on the principle of information visualisation for effective representation and more particularly to enhance rules interestingness measures.

3. **Interactive rules visualisation**

   We propose a methodology for the interactive visualisation of association rules: IUCEAR (Interactive User-Centred Exploration of Association Rules) that is intended to facilitate the user task when facing large sets of rules, taking into account his/her cognitive capabilities. In this methodology, the user builds himself/herself a reference rule which will be exploited by local algorithms in order to recommend better rules based on the reference rule. Then, the user explores successively a small set of rules using interactive visualisation related with suitable interaction operators. This approach is based on the principles of information cognitive processing.

4. **Local extraction of association rules**

   We develop specific constraint-based algorithms for local association rules extraction. These algorithms extract only the rules that our approach is considers interesting for the user. These algorithms use powerful constraints that significantly restrict the search space. Thus, they give the possibility to overcome the limits of exhaustive algorithms such as *Apriori* (the local algorithm extracts only a small sub set of rules at each user action). By exploring rules and changing constraints, the user may control both rules extraction and the post-processing of rules.

5. **The Virtual Reality visualisation tool IUCAREVis**

   IUCAREVis is a tool for the interactive visualisation of association rules. It implements the three previous approaches and allows rules set exploration, constraints modification, and the identification of relevant knowledge. IUCAREVis is based on an intuitive display in a virtual environment that supports multiple interaction methods.

Keywords: Association Rules Mining, Virtual Reality, Visualisation, Visual Data Mining, Interactive Rules Exploration.

Acknowledgments

The following dissertation, while an individual work, benefited from the insights and direction of several people.

This thesis would not have been possible unless it was financially supported by *Pays de la Loire Region* of France; *MILES project* was in charge with the administration of my financial contract. Thus, I would like to thank the council of the Pays de la Loire Region for giving me the possibility to follow my dreams...

I owe my deepest gratitude to Mr. Fabrice Guille, Mr. Paul Richard, Mr. Fabien Picarougne and Mr. Julien Blanchard my PhD supervisors. Mr. Fabrice Guillet believed in me, guided me and gave me precious advices throughout this work. He gave me the possibility to get this far by always encouraging me to go further. Mr. Paul Richard co-supervised my PhD, he provided timely and instructive comments and evaluation of my work allowing me to progress in my research. Our discussions were both constructive and enlightening, and I thank him.

I wish to express my gratitude to Mr. Gilles Venturini and Mr. Mustapha Lebbah, for the honor that they made me by accepting to review my thesis and for all their constructive remarks that allowed me to improve my dissertation. I would like also to thank Mr. Colin De La Higuera and Ms. Hanene Hazzag, for making me the honor to accept being examinators.

I had the pleasure to work in the *COnnaissances et Décisions - KnOwledge and Decisions (KOD)* research team of Nantes-Atlantique Computer Science Laboratory (LINA UMR 6241), in the Computer Science Department of Ecole polytechnique of University of Nantes. I am grateful to Ms. Pascale Kuntz for giving me the great privilege of joining the research team that she pilots. She was always there bringing me priceless answers and advices. My colleagues were always sources of laughter, joy, and support. They made my days less harder than they seemed to be.

In all of the ups and downs that came my way during the PhD years, I knew that I had the support of my husband, I would like to thank him – he was always there, listening and encouraging me, and always understanding; thank you.

Without you, my family, I would be nothing.

Contents

List of Figures

List of Tables

Introduction

Context

The progress made in day's current technology allows computer systems to store very large amounts of data. Never before has data been stored in such large volumes as today (Keim 2002 [168]). The data are often automatically recorded by computer, even for each simple transaction of every day life, such as paying by credit card or using a mobile phone. The data are collected because people believe that they could potentially be advantageous for management or marketing purposes.

This accumulation of information in databases has motivated the development of a new research field: Knowledge Discovery in Databases (KDD) (Frawley *et al.* 1992 [105]) which is commonly defined as the extraction of potentially useful knowledge from data. KDD is an iterative process and requires interaction with the decision maker both to make choices (pre-processing methods, parameters for data mining algorithms, etc.) and to examine and validate the produced knowledge.

One of the most frequently-used data mining methods is: Association Rules. In cognitive science, several theories of knowledge representation are based on rules (Holland *et al.* 1986 [147]). Generally, the rules are of the following form: "if antecedent then consequence", noted $Antecedent \rightarrow Consequent$ where the antecedent and the consequence are conjunctions of attributes of the database and values that they should take. Association rules have the advantage of presenting knowledge explicitly which can be easily interpreted by an non-expert user. Association rules were initially introduced by Agrawal *et al.* 1993 [2] for discovering regularities between products in large scale databases recorded by supermarkets. It finds combinations of products that are often purchased together in a supermarket. For example, if a customer buys milk, then he/she probably also buys bread.

Since the *Apriori* algorithm proposed by Agrawal and Srikant 1994 [3] which is the first proposed algorithm for extracting association rules, many other algorithms have been presented over-time. These algorithms use two interestingness measures (support and confidence) to validate the extracted association rules. The extracted rules should be validated beyond a user-specified minimum support and above a user-specified minimum confidence level. The support measure is the proportion of transactions in the database that satisfies the antecedent and the consequent (for example 3% of customers buy milk and bread). The confidence measure is the proportion of transactions that verify the consequent among those that verify the antecedent (for

example 95% of customers who buy milk buy also bread). The association rules generation algorithm is usually separated into two steps. Firstly, a minimum support is applied to find all frequent itemsets in a database. Secondly, these frequent itemsets are used to form rules whose confidence is above the minimum confidence constraint.

Problematic

One of the characteristics of the association rules extraction algorithms is to be unsupervised; they do not require target items but consider all possible combinations of items for the antecedent and for the consequent.
This feature enhances the strength of association rules, since algorithms require no prior data knowledge. Association rules algorithms can discover rules that the user considers interesting even if they consist of combinations of attributes which he/she would not have necessarily thought of. However, the same feature also constitutes the main limitation of association rules algorithms, since the amount of generated rules by an algorithm increases exponentially according to the number of attributes in the database. In practice, the volume of generated rules is prohibitive, reaching hundreds of thousands of rules.

To handle the large quantity of rules produced by the data mining algorithms, different solutions have been proposed to assist the user finding interesting rules :

- interestingness measures have been developed to evaluate rules in different perspectives (Tan and Kumar 2000 [209], Geng and Hamilton 2006 [117], Guillet and Hamilton 2007 [131]). They allow the user to identify and reject low-quality rules, and also to order acceptable rules from the best to the worst.

- redundancy rule reduction is proposed to reduce the number of generated rules by discarding redundant or nearly redundant rules. If a set of rules means the same thing or describes the same database rows, then the most general rule may be preserved.

- the interactive exploration of rules (Fule and Roddick 2004 [115], Yamamoto *et al.* 2009 [72], Blanchard *et al.* 2007 [29]) is proposed to assist the user in finding interesting knowledge in the post-precessing step. Several software applications have been developed with this in mind.

- visualisation can be effective for the user by displaying visual representations of rules (Bruzzese and Davino 2008 [51], Couturier *et al.* 2007 [80], Beale 2007 [20], Techapichetvanich and Datta 2005 [271]). This facilitates the understanding and accelerates rules ownership by the user.

Despite these this progress, several issues still remain. Firstly, the visual representations for association rules post-processing are generally not interactive. Thus,

they are used as complementary tools to present results in a more understandable form, but do not allow the user to look for interesting rules or to adjust the parameters of the association rules extraction algorithms. In addition, interactivity in the association rules post-processing is often poor. Thus, interactions are not fully adapted to the interactive character of the post-processing approach, and in particular do not take into account the special status of data. To better consider the user's interactivity needs, KDD processes must not only be viewed from the data mining perspective but also from the user perspective such as in user-centred systems for decision support. Finally, most of the approaches are massively limited to the "support/confidence" framework. Alone, these two measures does not allow the user to evaluate the pertinence of an association rule. Furthermore, the displayed rules interestingness measures are weakly enhanced although they are crucial indicators for post-processing. On the other hand, all proposed representations for association rules visualisation have been developed to represent association rules without paying attention to the relations between attributes which make up the antecedent and the consequent, and the contribution of these to the rule, in spite of the fact that the association rule attributes may be more informative than the rule itself (Freitas 1998 [109]).

The need for visualisation and interaction

Information visualisation can help the user deal with large amount of data by representing it in visual form to improve cognition for acquisition and the use of new knowledge. Unlike scientific visualisation which is constructed from measured or simulation data representing objects associated with phenomena from the physical world, information visualisation is therefore a visual representation of information that has no obvious representation. Visualisation improves cognitive tasks since it is based on the perceptual abilities of the human visual system. Without considering cognitive psychology, it can be said that visualisation improves the following attributes (Card *et al.* 1999 [60], Ceglar *et al.* 2003 [63], Ware 2004 [287], Ward *et al.* 2010 [286]):

- identification of similarities;
- identification of singularities;
- identification of structures;
- memorisation.

In particular, some visual information such as, position, size or colour are processed unconsciously and very rapidly by the human brain (Card *et al.* 1999 [60], Bertin 1984 [24]). A human can instantly and accurately determine the most populous city among a hundred other cities on a histogram. Executing the same task from textual information requires much more time and effort. With the arrival of the computer, visualisation has become dynamic; it is now an interactive activity. Visual Data

Mining (VDM) (Michalski *et al.* 1998 [195]), has been defined by Ankerst 2001 [10] as "a step in the Data Mining process that utilises visualisation as a communication channel between the computer and the user to produce novel and interpretable patterns". VDM is an approach dedicated to interactive exploration and knowledge discovery that is built on the extensive use of visual computing (Gross 1994 [129]). In his ecological approach to visual perception Gibson 1996 [120] established that perception is inseparable from the action. Thus, VDM studies do not only produce the best representations to improve cognition, but also the best interaction to implement these representations.

In 2D space, VDM has been studied extensively. More recently, hardware progress has led to the development of real-time interactive 3D data representation and immersive Virtual Reality (VR) techniques. VR lies at the intersection of several disciplines such as computer graphics, computer aided design (CAD), simulation and collaborative work. It uses hardware devices and multimodal interaction techniques to immerse one or more users in a Virtual Environment (VE). These techniques are based on human natural expression, action and perception abilities (Burdea and Coiffet 1993 [54] Fuchs *et al.* 2003 [112]). Thus, aesthetically appealing element inclusion, such as 3D graphics and animation, increases the intuitiveness and memorability of visualisation. Also, it makes the perception of the human visual system easier (Spence 1990 [255], Brath *et al.* 2005 [47]). In addition VR is flexible, in the sense that it allows different representations of the same data to better accommodate different human perception preferences. In other words, VR allows for the construction of different visual representations of the same underlying information, but with a different look. Thus, the user can perceive the same information in different ways. On the other hand, VR also allows the user to be immersed and thereby provides a way to navigate through the data and manipulate them from inside. VR hence creates a living experience in which the user is not a passive observer, but an actor who is part of the world, in fact, part of the information itself. In VR, the user may see the data sets as a whole, and/or focus on specific details or portions of the data. Finally, in order to interact with a virtual world, no mathematical knowledge is required, only minimal computer skills (Valdes 2003 [283]).

In this context, the use of VR techniques is very relevant: it allows the user to quickly view and select rules that seem interesting. The selection can be made intuitively, via the use of a gestural interface such as tracking devices or a dataglove in immersive configurations, or by mouse clicks in desktop configurations. The advantage of immersive configurations, (large screen and stereoscopic viewing) is that it improves data visualisation and may support multi-user work. However, VR techniques are still relatively little used in the context of VDM. We believe that this technological and scientific approach has a high potential to efficiently assist the user in analytical tasks.

Contribution

The contribution of the thesis is divided into 5 topics. Firstly, we elaborate an overview of interaction techniques and 3D representations for data mining. Then, we propose a new association rule metaphor to represent items that make up the antecedent and the consequent of an association rule. In addition, we propose a new approach to assist the user in the post-processing of association rules: interactive rules visualisation. Then, we adapt the extraction rules to the interactive nature of post-processing by developing specific algorithms for local association rules extraction. Finally, we implement our approach in the Virtual Reality visualisation tool we call IUCAREVis (Interactive User-Centered Association Rules Exploration and Visualisation).

1. **Classification for Visual Data Mining based on both 3D representations and interaction techniques**

   We present and discuss the concepts of visualisation and visual data mining. Then, we present 3D representation and interaction techniques in the context of data mining. Furthermore, we propose a new classification for VDM, based on both 3D representations and interaction techniques. Such a classification may help the user choose a visual representation and an interaction technique for a given application. This study allows us to identify limitations of the knowledge visualisation approaches proposed in the literature.

2. **Metaphor for association rule representation**

   We propose a new visualisation metaphor for association rules. This new metaphor takes into account more accurately the attributes of the antecedent and the consequent, the contribution of each one to the rule, and their correlations. This metaphor is based on the principle of information visualisation for effective representation and more particularly to enhance rules interestingness measures.

3. **Interactive rules visualisation**

   We propose a methodology for the interactive visualisation of association rules: IUCEAR (Interactive User-Centred Exploration of Association Rules) that is intended to facilitate the user task when facing large sets of rules, taking into account his/her cognitive capabilities. In this methodology, the user builds himself/herself a reference rule which will be exploited by local algorithms in order to recommend better rules based on the reference rule. Then, the user explores successively a small set of rules using interactive visualisation related with suitable interaction operators. This approach is based on the principles of information cognitive processing.

4. **Local extraction of association rules**

   We develop specific constraint-based algorithms for local association rules extraction. These algorithms extract only the rules that our approach is considers interesting for the user. These algorithms use powerful constraints that significantly restrict the search space. Thus, they give the possibility to overcome the limits of exhaustive algorithms such as *Apriori* (the local algorithm extracts only a small sub set of rules at each user action). By exploring rules and changing constraints, the user may control both rules extraction and the post-processing of rules.

5. **The Virtual Reality visualisation tool IUCAREVis**

   IUCAREVis is a tool for the interactive visualisation of association rules. It implements the three previous approaches and allows rules set exploration, constraints modification, and the identification of relevant knowledge. IUCAREVis is based on an intuitive display in a virtual environment that supports multiple interaction methods.

Thesis Organisation

This manuscript is organised as follows:

Chapter 2 is concerned with Knowledge Discovery in Databases (KDD), and more precisely by Association Rule Mining techniques. It provides formal definitions and considers the limits of the classic algorithms for association rules generation and the proposed solutions found in the literature.

Chapter 3 introduces the visualisation and the VDM. We describe 3D representation and interaction techniques for VDM. Then, we present a new classification of visualisation tools in data mining, regardless of the mining method used – pre-processing methods, post-processing methods (association rules, clustering, classification, etc.)

Chapter 4 provides a detailed presentation of virtual reality (VR) and virtual environments (VEs). We presents and analyses the various interaction devices and interfaces commonly used in VR. In addition, we review existing 3D interaction techniques and metaphors used in VR applications. Then, we propose a classification of hardware configurations and visual displays enabling user immersion in VEs. Finally, we present a comparison between 2D, 3D and virtual reality techniques in the context of information visualisation and VDM.

Chapter 5 is dedicated to the post-processing IUCARE approach and IUCAREVis tool; we describe the IUCARE methodology with reference to the principle of information visualisation and cognitive principles of information processing. We present

the visualisation metaphor used to represent association rules, basic choices, and a validation study. We also present IUCAREVis features that have been achieved, and describe their implementation in detail.

Chapter 6 provides the association rules local mining algorithms. It present the architecture of IUCAREVis and discusses its choices that we made during the development. Also, it details the interaction techniques proposed in IUCAREVis.

Chapter 7 presents the conclusion of our contribution and give some proposals for future work.

1

Knowledge Discovery in Databases and Association Rules

Contents

1.1 Introduction

Knowledge Discovery in Databases (KDD) is the process of extracting interesting patterns from data. The KDD process is commonly defined in three successive stages: Data Pre-Processing; Data Mining; and finally Post-Processing. In Data Mining, different techniques can be applied among which association rule mining is one of the most popular.

The association rule mining method proposes the discovery of knowledge in the form of *IF Antecedent THEN Consequent* noted $Antecedent \rightarrow Consequent$. In an association rule, the antecedent and the consequent are conjunctions of attributes in a database. More particularly, an association rule $Antecedent \rightarrow Consequent$ expresses the implicative tendency between the two conjunctions of attributes – from the antecedent toward the consequent.

The main advantage of the association rule mining technique is the extraction of comprehensible knowledge. On the other hand, the main disadvantage of this method is the volume of rules generated which often greatly exceeds the size of the database. Typically only a small fraction of that large volume of rules is of any interest to the user who is very often overwhelmed by the massive amount of rules. The cognitive processing of thousands of rules takes much more time then generating them even by a less efficient tool. Imielinski *et al.* 1998 [152] believe that the main challenge facing association rule mining is what to do with the rules after having generated them.

To increase the efficiency of the rule generation process (to reduce the number of discovered rules) several methods have been proposed in the literature. Firstly, different algorithms have been developed to reduce the number of generated rules. Secondly, several methods have been proposed to help the user to filter the algorithm results. In this chapter we will look at mainly three of these methods: interestingness measures, redundancy rule reduction, and interactive rule post-processing.

This chapter starts with a brief presentation of Knowledge Discovery in Databases. The second part is dedicated to association rule mining, definitions and notations. The third part presents algorithms for association rule extraction. Finally, the forth part presents the problematic of association rule techniques and the solutions proposed in the literature to fulfil it.

1.2 Knowledge Discovery in Databases

Knowledge Discovery in Databases (KDD) was defined by Frawley *et al.* 1992 [105], and revised by Fayyad *et al.* 1996 [101], as the *non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.*

KDD is a multi-disciplinary field, being integrated in areas such as artificial intelligence, machine learning, neural networks, data bases, information retrieval and data visualisation. Furthermore, the KDD process is applied in various research fields. In the 1990s, there were only a few examples of knowledge discovery in real data. Nowadays, more and more domains benefit from the utilisation of KDD techniques, such as medicine, finance, agriculture, social work, marketing, military, and many others.

The KDD process aims at the extraction of hidden predictive information from large databases. KDD methods browse databases to find hidden knowledge that experts may miss because it is outside their expectations. Most companies already collect and refine massive quantities of data and KDD is becoming an increasingly important technique to transform this data into knowledge. Thus, KDD is commonly used in a wide range of domains, and is characterised as being a *non-trivial* process because it can decide whether the results are interesting enough to the user. This defines the degree of evaluation autonomy.

Fayyad *et al.* 1996 [101] defined four notions to characterise the extracted patterns: *validity*, *novelty*, *usefulness* and *comprehension* by users. Firstly, the extracted patterns should be *valid* for new data with some degree of certainty described by a set of interestingness measures (e.g. confidence measure for association rules). Secondly, the *novelty* of patterns can be measured with respect to previous or expected values, or knowledge. Next, the patterns should be *useful* to the user which means that useful patterns can help the user to take beneficial decisions. The *usefulness* characteristic considers that knowledge is externally significant, unexpected, non-trivial, and actionable. Lastly, the extracted patterns should be *comprehensible* by analysers, who should be able to use them easily to take decisions.

At least two of the four characteristics (novelty and usefulness) require a direct user implication in the KDD process which explains the need for interactivity during the KDD process. Figure 1.1 presents the main KDD steps: Data Pre-Processing, Data Mining, and Post-Processing of discovered patterns (Fayyad *et al.* 1996 [281]).

1.2.1 Data Pre-Processing

This step consists of three tasks: *Data Cleaning*, *Data Integration* and *Data Validation*.

- Data Cleaning
Real-life data contains noise and missing values which are considered inconsistent. Applying the KDD process over this data may extract data of poor reliability. The *Data Cleaning* step consists of detecting and correcting (or removing) inaccurate and inconsistent data from the database. Generally, automatic systems based on statistical methods are needed to analyse the data and to replace missing or incorrect data by one or more plausible values. For example, if values are missing for some attributes,

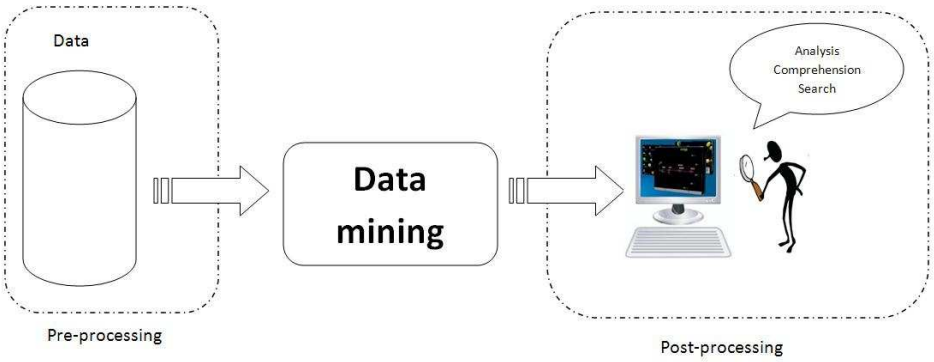


FIGURE 1.1: An overview of the KDD process.

this step allows them to be computed by using heuristics. Another example is when some values are inserted into the data by error. In this case, a set of methods can be applied in order to determine which values are incorrect.

- Data Integration
Data Integration is used to collect data from multiple sources and to provide users with a unified view of these data. The resulting database can presents incoherence and the *Data Integration* step proposes solutions for this kind of problem. A valuable example is redundancy. If an attribute A can be determined from another attribute B, then A is redundant compared to B. Another type of redundancy is the existence of two attributes from different sources with different names, but which represent the same information. One of them should be removed from the final data.

- Data Validation
The goal of *Data Cleaning* and *Data Integration* is to generate a database which contains modified data. This data makes future analysis processes easier. Once the database has been created, *Data Validation* is used to achieve two goals. The first is to verify if the database was well developed during the *Data Cleaning* and the *Data Integration* phases; if needed, data can be re-cleaned. The second goal of this step is to transform (or to reduce) the data allowing the KDD process to apply a knowledge discovery technique. Data Mining can only uncover patterns already present in the data. The target dataset must be large enough to contain these patterns while remaining concise enough to be mined within an acceptable time frame.

1.2.2 Data Mining

Data Mining step is central in the KDD process. *Data Mining* consists of applying data analysis and discovery algorithms to produce knowledge. Four main classes of tasks have been developed in the literature in order to extract interesting patterns.

- Classification
Classification builds a model in order to map each datum into one of several predefined classes. The classification is composed of two phases. The first one is the learning phase – the description of a set of classification rules called a learning model. The second phase is classification – verifying the precision of the classification rules generated during the first phase by using test data. For instance, an e-mail program might attempt to classify an email as legitimate or spam. The main classification techniques are: *Decision Trees*, *Bayesian Classification*, and *Neural Networks*.

- Clustering
The clustering technique identifies a finite set of classes or clusters which describe data. This method partitions the data into classes in such a way that the intraclass similarity be maximised and the interclass similarity be minimised. In a first step, all the adequate classes are discovered, then the data are classified into those classes. Compare to classification, the classes are not known from the beginning, they are discovered using a set of observations. Different methods of clustering have been developed, among which the *K-means* method.

- Regression analysis
Regression analysis is the oldest and best-known statistical technique used in Data Mining. Basically, regression analysis takes a numerical dataset and develops a mathematical formula that fits the data. To create a regression model, a specific parameters value – which minimise the measure of the error, should be found. A large body of techniques for carrying out regression analysis has been developed. Familiar methods such as linear regression and least squares (Legendre 1805 [179]) are presented.

- Association Rules
This technique aims to discover interesting rules from which new knowledge can be derived. Finding association rules consists of finding regularities in data by searching for relationships among variables (Piatetsky-Shapiro and Frawley 1991 [219]).
Association rules is a frequent implications in data of the type *IF X THEN Y*; X and Y represent respectively the antecedent and the consequent. The association rule mining system's role is to facilitate the discovery and to enable the easy exploitation and comprehension of results by humans. *Association rules* have been found to be useful in many domains such as business, medicine, etc. For example a supermarket might gather data on customer purchasing habits which aims to predict user behaviour. Using association rule mining, the supermarket can determine which products are frequently bought together and uses this information for marketing purposes.

1.2.3 Post-processing of Discovered Patterns

Usually called *post-processing* (Baesens *et al.* 2000 [15]) or *post-mining*, this phase is the final step of the KDD process. The *Data Mining* algorithm discovers a list of *patterns* with a given level of interest; the purpose of this step is to verify if the produced *patterns* can be considered as a *knowledge*. Not all patterns found by the

data mining algorithms are necessarily valid.

The notion of interest or *interestingness* was defined by Silberschatz and Tuzhilin, 1996 [248] to describe the interest of a pattern. This notion is presented as a general measure over nine characteristics: *conciseness*, *coverage*, *reliability*, *peculiarity*, *diversity*, *novelty* , *surprisingness*, *utility* and *actionnability*. Thus, a pattern which meets one or more criteria is consider as *interesting* and can be validated as a *knowledge*.

In most cases, it is the user who evaluates the discovered patterns, i.e. to determine if the extracted pattern is interesting or not, in the *post-processing* step. Several user-driven methods and statistical database-oriented methods are available to assist the user in this task. For example, it is common for the *classification* algorithms to find patterns in the training set which are not present in the general data set; this is called overfitting. In this case, it is important that the user be able to eliminate them. For instance, a *Data Mining* algorithm trying to distinguish spam from legitimate e-mails would be trained on a training set of sample e-mails. Once trained, the learned patterns would be applied to the test set of e-mails which had not been used for training. The accuracy of these patterns can then be measured by how many e-mails were correctly classified. Another method of pattern evaluation and validation is visualisation, which is related to the model of extracted patterns (see Chapter 3.3).

1.3 Association Rule Mining

Association rule mining, the task of finding correlations between attributes in a dataset, has received considerable attention, particularly since the publication of the *AIS* and *Apriori* algorithm by Agrawal et *al.* 1993 [2] and Agrawal and Srikant 1994 [3]. Initial research was largely motivated by the analysis of market data. The result of these algorithms allows companies to better understand purchasing behaviour, and, as a result, better target market audiences. Association rule mining has since been applied to many different domains – all areas in which relationships among objects provide useful knowledge. In this section, we present association rules and formally describe the main notions, since they are at the very foundation of this thesis.

1.3.1 Presentation

Research in association rules was first motivated by the analysis of market basket data. But, how could a set of shopping tickets produce some modifications in supermarket layout?

In a first analysis of shopping basket data of a supermarket searching for purchasing behaviour, the decision-maker found a strong correlation between two products A and B, of the form $X \rightarrow Y$, where X (antecedent) and Y (consequent) are non-intersecting sets of attributes. For instance, $milk \rightarrow bread$ is an association rule saying that when milk is purchased, bread is likely to be purchased as well. Such extracted information can be used to make decisions about marketing activities such

as promotional pricing or product placement. In our example, we could more efficiently target the marketing of bread through marketing to those clients that purchase milk but not bread. Increasingly, association rules are currently employed in many application areas including Web use pattern analysis (Srivastava et *al.* 2000 [260]), intrusion detection (Luo and Bridges 2000 [186]) and bioinformatics (Creighton and Hanash 2003 [81]).

1.3.2 Terminology and Annotations

In general, the association rule mining technique is applied over a database $D = \{I, T\}$. Let us consider $I = \{i_1, i_2, \ldots, i_m\}$ a set of m binary attributes, called *items*. Let $T = \{t_1, t_2, \ldots, t_n\}$ be a set of n transactions, where each transaction t_i represents a binary vector, with $t_i[k] = 1$ if t_i contains the item i_k, and $t_i[k] = 0$ otherwise. A unique identifier is associated to each transaction, called *TID*. Let X be a set of items in I. A transaction t_i *satisfies* X if all the items of X exist also in t_i, formally, we can say that $\forall i_k \in X,\ t_i[k] = 1$. In conclusion, a transaction t_i can be viewed as a subset of I, $t_i \subseteq I$.

Definition 1.3.1

An itemset $X = \{i_1, i_2, \ldots, i_k\}$ is a set of items $X \subseteq I$. We can denote the itemset X by $i_1, i_2, \ldots, i_k$, the comma being used as a conjunction, but most commonly it is denoted by $i_1 i_2 \ldots i_k$, omitting the commas.

Example 1.3.2 Let us consider a sample supermarket transaction dataset:

Tuple	Milk	Bread	Eggs
1	1	0	1
2	1	1	0
3	1	1	1
4	1	1	1
5	0	0	1

Table 1.1: Supermarket transaction dataset

Suppose that D is the transaction table shown in Table 1.3.2, which describes five transactions (rows) involving three items: milk, bread, and eggs. In the table, 1 signifies that the item occurs in the transaction and 0 means that it does not.

The Tuple 4 = *Milk Bread Eggs* (or *Milk, Bread, Eggs*) is an itemset composed by three items: *Milk*, *Bread* and *Eggs*.

Definition 1.3.3

An itemset X is a k-itemset if X is an itemset $X \subseteq I$ and if it contains k items: $|X| = k$.

Example 1.3.4 The itemset *Milk*, *Bread*, *Eggs* is a *3*-itemset.

Definition 1.3.5

Let $X \subseteq I$ and $t_i \in T$. t(X) is the set of all transactions which contain the itemset X. t(X) is defined by:

$$t : \mathcal{P}(I) \to T,\ t(X) = \{t_i \in T \mid X \subseteq t_i\}.$$

In a first attempt, an association rule was defined as an implication of the form $X \to y_i$, where X is an itemset $X \subseteq I$ and y_i is an item $y_i \in I$ with $\{y_i\} \cap X = \emptyset$ Agrawal et *al.* 1993 [2].

Later, the definition was extended to an implication of the form $X \to Y$, where X and Y are itemsets and $X \cap Y = \emptyset$ (Agrawal and Srikant 1994 [3]). The former, X, is called the antecedent of the rule, and the latter, Y, is called the consequent of the rule.

A rule $X \to Y$ is described by two important statistical factors: support and confidence.

Definition 1.3.6

The support of an association rule is defined as the support of the itemset created by the union of the antecedent and the consequent of the rule

$$supp(X \to Y) = supp(X \cup Y) = |t(X \cup Y)| = \frac{P(X \cup Y)}{T}.$$

The support presents the proportion of transactions in the data set which contains both X and Y. If $supp(X \to Y) = s$, s% of transactions contain the itemset $X \cup Y$.

Definition 1.3.7

The confidence of an association rule is defined as the probability that a transaction containing Y also contains X. Therefore, the confidence is the ratio (c%) of the number of transactions that contain X, as well as Y:

$$confidence(X \to Y) = \frac{supp(X \to Y)}{supp(X)} = \frac{supp(X \cup Y)}{supp(X)}.$$

In most cases, association rules extraction algorithms seek to satisfy a user-specified minimum support threshold and a user-specified minimum confidence threshold at the same time. The association rule generation is always a two-step process: firstly the minimum support threshold is applied to find all frequent itemsets in a database, then these frequent itemsets and the minimum confidence threshold constraint are used to validate the extracted rules.

Definition 1.3.8

We note the minimum support threshold provided by the user as minSupp, and

the minimum confidence threshold as $minConf$. An association rule $X \rightarrow Y$ is valid if:

- *the support of the rule is greater than minSupp: $supp(X \rightarrow Y)$ $minSupp$;*
- *the confidence of the rule is greater than $minConf$: $conf(X \rightarrow Y)$ $minConf$.*

Example 1.3.9 Let us consider a sample of supermarket transaction dataset shown in Table 1.3.2:
The association rule AR: $Milk \rightarrow Bread$ can be generated from D. The level of rule support is 60% because the combination of $Milk$ and $Bread$ occurs in three of the five transactions, and the confidence is 75% because $Bread$ occurs in three of the four transactions that contain $Milk$

Definition 1.3.10
The lift was firstly defined by Brin et al. 1997 [48] pointing out the importance of the correlation between the antecedent and the consequent.

The lift is defined as :

$$Lift(X \rightarrow Y) = \frac{P(X,Y)}{P(X)P(Y)} = \frac{supp(X \cup Y)}{supp(X)supp(Y)} = \frac{Confidence(X,Y)}{P(Y)}$$

The Lift measures the degree of deviation from an independent case. A rule is considered independent if X and Y are independent: $P(X \cup Y) = P(X)P(Y)$.

Let us compute the lift value in the case of independence:

$$lift(X \rightarrow Y) = \frac{P(X,Y)}{P(X)P(Y)} = \frac{P(X)P(Y)//independence\ case}{P(X)P(Y)} = 1.$$

Accordingly, the more that lift is greater than 1, the greater the interest of the rule.

Example 1.3.11 Let us consider a simple association rule $Milk \rightarrow Bread$ [C = 75%] – in 75% of cases, when we have $Milk$ in a supermarket basket, we also have $Bread$. The confidence evaluates the rule as being interesting. On the other hand, the lift value could prove the contrary. The result depends on the support of the $Bread$ item in the database. Two cases are possible:

- $supp(Bread) = 75\%$: alone, $Bread$ item appears in 75% of baskets. Thus, it is not surprising to have a confidence of 75%, because in reality, $Milk$ does not

increase its chances to be in a supermarket basket. In consequence, computing the lift as $Lift(AR) = 1$, indicate that there is no dependence between the two items $Milk$ and $Bread$. Therefore, the rule is not interesting.

- $supp(Bread)! = 75\%$: the more the support of $Bread$ is different from 75%, the more the rule is interesting.

Definition 1.3.12

The Information Gain was defined by Freitas 1999 [107] to evaluate the Information Gain of rule antecedent attributes. The Information Gain is defined as :

$$InfoGain(Ai) = Info(G) - Info(G|Ai)$$

$$Info(G) = -\sum_{j=1}^{n} Pr(Gj) log Pr(Gj)$$

$$Info(G|Ai) = \sum_{k=1}^{m} Pr(Aik)(-\sum_{j=1}^{n} Pr(Gj|Aik))$$

where :

- *n: the number of consequent attribute values;*
- *m: the number of values of the anticipation attribute A_i;*
- *InfoGain(A_i): the information gain of each attribute A_i in the rule antecedent;*
- *Info(G): the information of the rule consequent.*
- *$Info(G|A_i)$: the information of the consequent attributes G given the antecedent attribute A_i, A_{ij} denotes the j-th value of attribute A_i;*
- *G_j: the j-th value of the consequent attribute G;*
- *Pr(X): the probability of X;*
- *$Pr(X|Y)$: the conditional probability of X given Y.*

The *Information Gain* measure can be positive or negative. An item with high positive *Information Gain* is considered as a good predictor for the rule consequence. An item with high negative *Information Gain* is considered as a bad one and should be removed from the association rule. From a rule interest perspective, the user already knows the most important attributes for its field, and the rules containing these items may not be very interesting. At the same time, a rule including attributes with low or negative information gain (logically irrelevant for the association rule consequence) can surprise the user in cases where attribute correlation can make an irrelevant item into a relevant one.

Example 1.3.13 Let us consider a simple association rule $Milk, Bread \rightarrow Eggs$. Lets suppose that the *Information Gain* of Milk and the *Information Gain* of Bread are:

$InfoGain(Milk) = -0.7$

$InfoGain(Bread) = 0.34$

We can conclude that $Bread$ is more interesting than $Milk$ which has a negative *Information Gain*. Each time the consumer purchases $Bread$, he/she purchases $Eggs$ but he/she does not purchase $Milk$. The *Information Gain* indicates that the implication $Milk \rightarrow Eggs$ is not valid.

1.4 Algorithms for Association Rule Extraction

Association mining analysis is a two part process. Firstly the identification of sets of items or itemsets within the dataset. Secondly, the rule generation from these itemsets. As the complexity of the itemset identification is significantly greater than that of rule generation, the majority of research in association rule extraction algorithms has focused on the efficient discovery of itemsets. Given n distinct items (n = $\|I\|$)within the search space, there are $2^n - 1$ (excluding the empty set which is not a valid itemset) possible combinations of items to explore. This is illustrated in Figure 1.2 which shows the search space lattice resulting from I = Milk, Bread, Eggs, Apples, Pears. Most of the time n is large, therefore naive exploration techniques are often difficult to solve.

Since the exhaustive reference algorithm proposed by Agrawal and Srikant 1994 [3], called *Apriori*, many algorithms inspired by *Apriori* have been proposed to efficiently extract association rules. In parallel, many constraint-based algorithms have been developed to extract association rules with constraints other than support and confidence. To summarise, relevant research can be organised into two groups of algorithms:

- Exhaustive algorithms
- Constraint-based algorithms

1.4.1 Exhaustive Algorithms

Exhaustive algorithms for association rule extraction all run on the same deterministic task: given a minimum support threshold and a minimum confidence threshold, they produce all rules that have support above the threshold (generality constraint) and a confidence above the threshold (validity constraint). Many adaptations and generalisations of association rules have been also studied. The main ones are: the

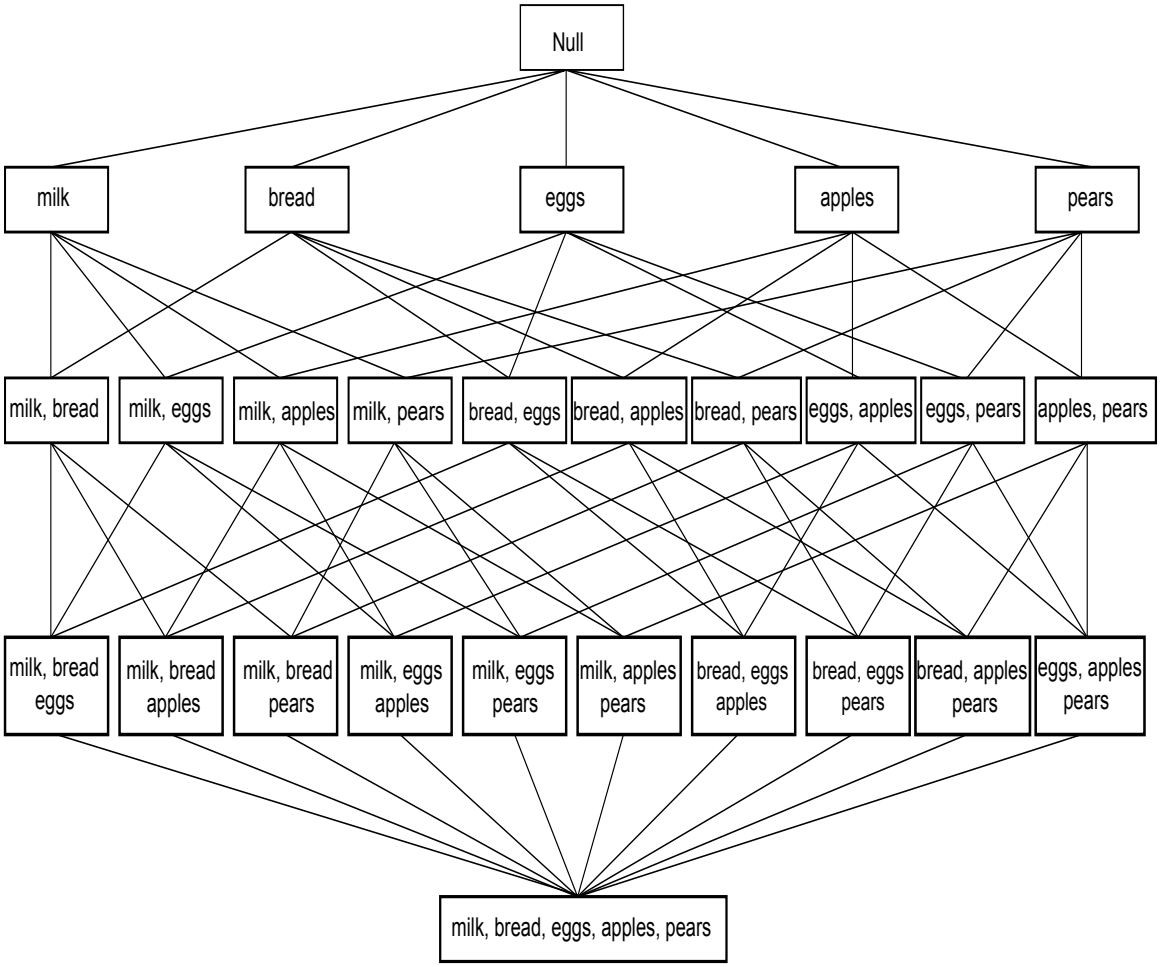


FIGURE 1.2: Search space lattice.

numeric association rules –involving quantitative variables (Srikant and Agrawal 1996 [257], Fukuda *et al.* 2001 [114]), the generalised association rules –to operate a hierarchy of concepts (Srikant and Agrawal 1997 [259], Han and Fu 1995 [135]), a nd the sequential patterns extracted from temporal data (Srikant and Agrawal 1996 [258], Mannila *et al.* 1997 [191], Zaki 2001 [306]).

Association rule extraction algorithms are often decomposed into two separate tasks:

- discover all frequent itemsets having support above a user-defined threshold $minSupp$.
- generate rules from these frequent itemsets having confidence above a user-defined threshold $minConf$.

Differences in performance between the different exhaustive algorithms depend mainly on the first task (Ng *et al.* 1998 [207]). The identification of valid itemsets is computationally expensive, because it requires the consideration of all combinations of distinct items in I (or $2^n - 1$ subset). The search space growth is exponential as n increases. Therefore, it is the first step that requires maximum efforts to optimise the association rule extraction algorithm. Itemset identification research thus focuses on reducing the number of passes over the data and on constraining exploration. The second task (rule generation) is less expensive. Nevertheless, there are two major problems with association rules generation:

- too many rules are generated (rule quantity problem).
- not all the rules are interesting (rule quality problem).

Both problems are not entirely independent. For example, knowledge about the quality of a rule can be used to reduce the number of generated rules.

1.4.1.1 *Apriori* – Classical Association Rule Mining

The fundamental exhaustive algorithm is *Apriori* which was designed by Agrawal and Srikant 1994 [3]. To generate frequent itemsets the *Apriori* algorithm uses the bottom-up, breadth-first method. This algorithm takes advantage of the downward closure property (also called anti-monotonic) of support to reduce the search space of the frequent itemset extraction: if an itemset is not frequent then any of it super-itemsets is frequent.

The *Apriori* algorithm has two main parts (i)frequent itemset generation and (ii) association rule generation.

Table 1.4.1.1 presents a frequent itemset generation task of *Apriori* which is carried out level by level. The set of candidates L_1 is formed by the set of items I, given k = 1, otherwise it is based on generating-itemset function involving members of L_{k-1}. More precisely, The algorithm gradually generates the set of itemsets from *1-itemsets* to *k-itemsets*. In the first pass over the data (line 1 in the algorithm), support for the *1-itemsets* is computed in order to select only the frequent ones. In the next steps (lines 2 to 10), the algorithm starts from the *(k-1)-itemsets* and uses the downward closure property to generate *k-itemsets*.

Thus, the function *generating-itemset* (line 3) generates new potentially frequent *k-itemsets* from the frequent *(k-1)-itemsets* already generated in the previous step. Potentially frequent itemsets are called *candidates*. During a new pass over the data, the support of each candidate is computed (lines 4 to 8). Then frequent candidates, that have support above the threshold, are validated.

Input: Database D
Output: L: a set of couple (I, sp(I)) when I is an itemset and sp(I) its support

1. $L_1 = \{1\text{-itemsets}\}$
2. **forall** $(k = 2; L_{k-1} \neq \emptyset; k++)$ **do begin**
3. C_k = generating-itemset(L_{k-1})
4. **forall** *transactions* $t \in D$ **do begin**
5. $C_t = subset(C_k, t)$
6. **forall** *candidates* $c \in C_t$ **do**
7. $c.count++$
8. **endfor**
9. $L_k = \{c \in C_k \mid c.count \geq minsup\}$
10. **endfor**

generating-itemset(L_{k-1})
12. **forall** *itemsets* $c \in C_k$ **do begin**
13. **forall** (k-1)-subsets s of c **do begin**
14. **if** $(s \in L_{k-1})$ **then**
15. **delete** c from C_k
16. **endfor**
17. **endfor**

18. **return** L

TABLE 1.2: Frequent itemset generation in an *Apriori* algorithm (Agrawal and Srikant 1994 [3]).

The *Apriori* algorithm has the particularity of using a support counting method. The function *subset* (line 5) receives the set of candidates and a transaction t of the database and returns the set of candidates satisfying the transaction. In line 7 the support of each candidate is increased. In line 9, the frequent *k-itemsets* are selected and they become the entry for the next step of the algorithm. The algorithm ends when no frequent itemset is generated.

Example 1.4.1 Let us consider a sample of the supermarket transaction database (Table 1.4.1) and a minimum support threshold of 50%.

In Figure 1.3, we present the process of generating frequent itemsets using by the *Apriori* algorithm. The algorithm starts with an empty list of candidates, and, during the first pass, all *1*-itemsets are generated. Only the itemsets satisfying the support constraint (50%) become candidates (black). Therefore, the itemsets that

Tuple	Transaction
1	Milk, Bread, Eggs, Apples
2	Milk, Bread, Apples
3	Bread, Eggs
4	Milk, Bread, Eggs, Pears

TABLE 1.3: Supermarket database sample for the *Apriori* algorithm example.

have support above the threshold (50%) are eliminated (red). The itemset $\{Pears\}$ with the support of 25% does not satisfy the support constraint and consequently is considered as a non-frequent item and, thus, is not kept as a candidate.

In the next passes, to reduce time execution the not-frequent itemsets are not computed. The k-itemsets which include an infrequent *(k-1)*-itemsets are considered as not frequent. For instance, the {*Bread, Pears*} {*Eggs, Pears*} {*Milk, Pears*} and {*Apples, Pears*} *2*-itemsets are not generated. The itemsets not containing the $\{Pears\}$ itemset are potentially frequent.

On the other hand, not all generated *2*-itemsets are frequent. For instance, the itemset {*Eggs, Apples*} is not frequent even though { *Apples*} and {*Eggs*} are frequent. A *(k+1)*-itemsets is frequent when all sub *(k-1)*-itemsets are frequent. For instance, the {*Milk, Bread, Eggs*} is frequent because the three *2*-itemsets composing it are frequent {*Milk, Bread*}, {*Milk, Eggs*} and {*Bread, Eggs*}. On the contrary, the {*Milk, Eggs, Apples*} itemset is not frequent because one of the three *2*-itemsets ({*Eggs, Apple*}) composing it is not frequent even though {*Milk, Eggs*} and {*Milk, Apples*} are frequent.

The second step of the *Apriori* algorithm is *rule generation*. This step aims to create association rules from the frequent itemsets generated in the first step. The algorithm is presented in Table 1.4.1.1.

The method used for rule extraction is very simple. Let us consider the set of frequent itemsets L. Considering $l_i \in L$, the method finds all subsets a of l_i, $a \subseteq l_i$, and proposes a set of rule candidates of the form $a \rightarrow (l_i - a)$. Only rules that have a confidence level above the threshold are generated.

In lines (1-2) the recursive procedure $generate - rules$ is called for each set of k-itemsets. $generate - rule$ generates recursively the sub-itemsets level by level (line 4) to produce rules which are further tested against the confidence level.

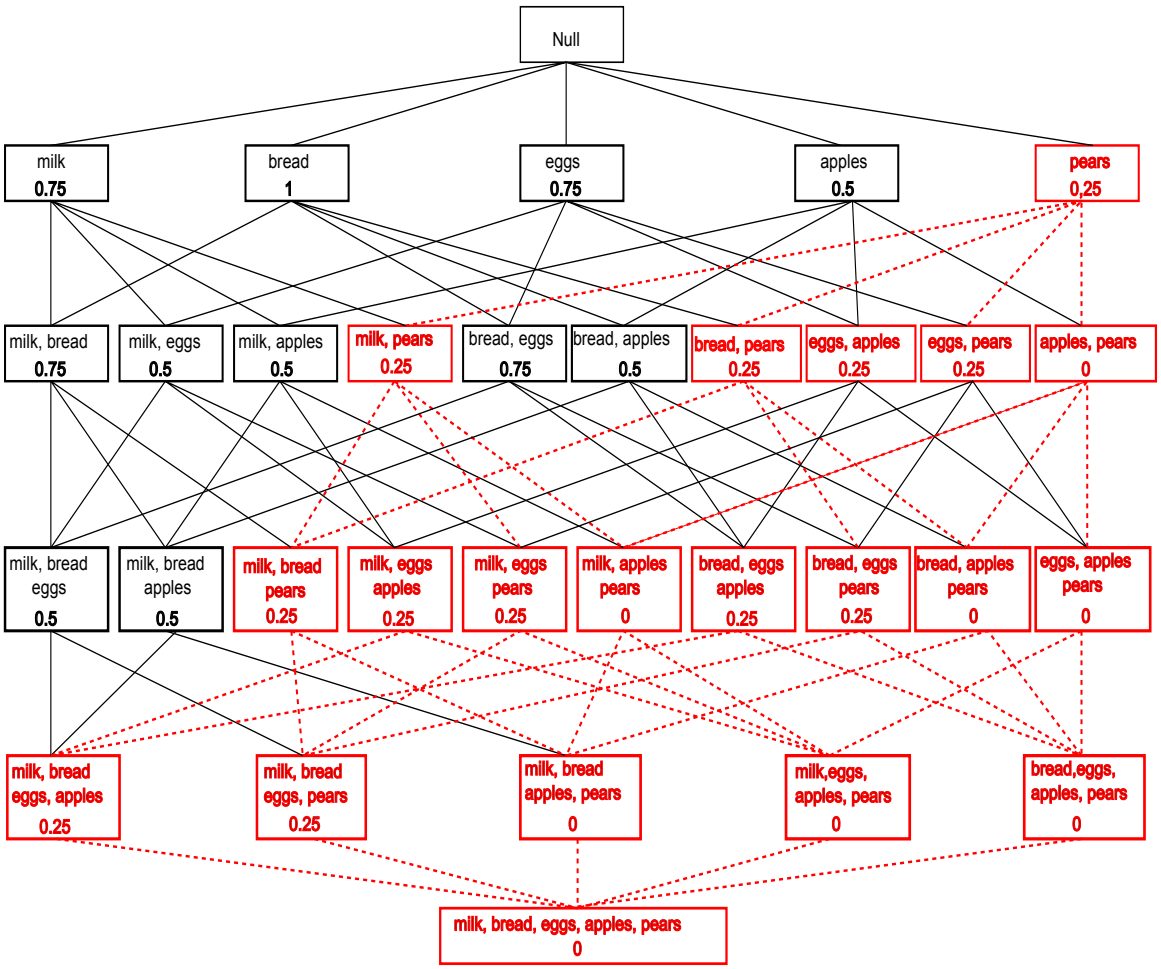


FIGURE 1.3: Tree of the frequent itemset generation.

Example 1.4.2 Let us consider the itemset $l_1 = \{Milk,\ Bread,\ Eggs\}$ [S = 50%]. six association rules can be extracted:

- *R1: Milk, Bread* $\rightarrow$ *Eggs* conf(R1) = 66%
- *R2: Milk, Eggs* $\rightarrow$ *Bread* conf(R2) = 100%
- *R3: Bread, Eggs* $\rightarrow$ *Milk* conf(R3) = 66%
- *R4: Milk* $\rightarrow$ *Bread,Eggs* conf(R4) = 66%
- *R5: Bread* $\rightarrow$ *Milk,Eggs* conf(R5) = 50%
- *R6: Eggs* $\rightarrow$ *Milk, Bread* conf(R6) = 66%

Input: Set of itemsets l
Output: Set of association rules *Rules*

1. **forall** itemsets l_k, $k \geq 2$ **do**
2. **call** $generate - rules(l_k, l_k)$;

3. procedure $generate - rules(l_k$: k-itemset, a_m: m-itemset)
4. $A = \{(m-1)$-itemsets $a_{m-1} \mid a_{m-1} \subset a_m \}$
5. **forall** $a_{m-1} \in A$ **do begin**
6. $conf = support(l_k)/support(a_{m-1})$
7. **if** $(conf \geq minConf)$ **then**
8. $R = a_{m-1} \Rightarrow (l_k - a_{m-1})$
9. **if** $(m - 1 > 1)$ **then**
10. **call** $generate - rules(l_k, a_{m-1})$
11. $Rules = Rules \cup R$
12. **return** *Rules*

TABLE 1.4: Rule generation step in *Apriori* algorithm [3].

We define the confidence threshold $minConf = 80\%$. Only $R2$ is generated by the algorithm.

The algorithm provides all frequent itemsets and their support which are necessary to calculate the rules for interestingness measures. For cases where the user is not interested by all itemsets and their support, algorithms for extracting maximal frequent itemsets have been developed (Bayardo and Roberto 1998 [18], Teusan *et al.* 2000 [272], Gouda and Zaki 2001 [123], Burdick 2001 [55]). Maximum frequent itemsets are frequent itemsets in which any of it's super-itemsets are frequent. They can easily find the frequent itemsets since all the frequent itemsets are composed of the set of maximal frequent itemsets and their sub-itemsets.

The efficiency of the *Apriori* algorithm depends on both the minimum support threshold and the studied data. From a qualitative point of view, we call sparse data (respectively dense) when the items present in the transactions are infrequent (respectively frequent) compared to the non-present items (proportion of 1 compared to 0). For a given support threshold:

- the more the data is sparse, the more the anti-monotonic property is effective to reduce the search space. Therefore, the algorithm can handle a large number of items;
- the more the data is dense, the less the anti-monotonic property is effective

to reduce the search space. Therefore, the algorithm cannot support a large number of items.

If the number of frequent itemsets generated by an algorithm makes it unusable, the only way to process the data is to increase the support threshold.

In this section, we saw that the *Apriori* algorithm is able to extract a set of association rules from a database. In this context, two problems concerning the quality of the algorithm emerge: *the rapidity* and *the efficiency*.

- *the rapidity*: deals with the capacity of the algorithm to generate the expected results in a reasonable time without using large quantities of resources. The frequent itemset generation step is the critical phase of the process. Itemset generation is an exponential problem, the search space to enumerate all the frequent itemsets is $2^n - 1$, where n is the number of items. Moreover, the algorithm makes several passes over data depending on the length of the generated itemsets. These tasks imply an exponential growth of the resources employed during the rule mining process and an considerable increase of the execution time. On the other hand, the rule generation step doesn't need new passes over the database, and hence, the execution time is not very high.
- *the efficiency*: deals with the capacity of the algorithm to produce interesting results. The main drawback of classical association rule mining techniques such as the *Apriori* algorithm is the huge number of produced rules which are quasi unusable by the user (millions of association rules can be extracted from large databases with a reduced support threshold). To address this shortfall, much research work has been carried out to reduce the number of extracted rules. In Section 1.5 we make a survey of rule number reduction methods.

1.4.1.2 Other algorithms

An important number of algorithms, based on *Apriori* in most cases, have been proposed with the aim of optimising frequent itemset generation step by introducing condensed representations, dataset partitioning, dataset pruning or dataset access reduction. Among the new algorithms we outline the most important ones: *FP-Growth* (Han and Pei 2000 [136]), *AprioriTID* (Agrawal and Srikant 1994 [3]), *Partition* (Savasere *et al.* 1995 [243]), and *Dynamic Itemset Couting (DIC)* (Brin *et al.* 1997 [48]).

To generate frequent itemsets, the major part of association rule extraction algorithms generate candidates and then check their support from database transactions. This is the most expensive step in exhaustive algorithms. The *Pattern Growth* algorithms have been introduced to eliminate the need for candidate generation and thus reduce the algorithms execution time. Instead of the candidate generation method, *Pattern Growth* algorithms use complex hyperstructures that contain representations

of the itemsets within the dataset.

The best known *Pattern Growth* algorithm is the *FP-Growth* algorithm introduced by Han and Pei 2000 [136]. Later, in 2005, Grahne and Zhu 2005 [124] developed the *FP-Growth** algorithm which improves the previous algorithm performance due to *FP-array*, a new data structure which allows the passes over the *FP-tree* to be improved.

FP-Growth is a memory-based algorithm and was developed to process dense data. The algorithm does not work directly on the data but on a condensed representation of it (*FP-tree*) to improve performance and efficiency of the frequent itemset step. The *FP-tree* is constructed by passing over all the itemsets in depth first (contrary to *Apriori*). The construction step needs two passes over the data. First, the algorithm constructs an *FP-tree* using the set of frequent singleton itemsets. Then, it maps each database transaction into a tree path.

After the construction step, the algorithm generates all frequent itemsets of various cardinalities from the *FP-tree* representation by successively concatenating those frequent singleton itemsets found in the tree path. If the *FP-tree* does not fit into the memory, recursive projections and partitioning are required to break such databases into smaller pieces. As a result there will be a corresponding performance overhead which will be similar to the *Apriori* algorithm. To overcome this limitation, several improvements of the *FP-tree* structure *FP-Growth* algorithm have been proposed (Cheung and Zaiane 2003 [69], Coenen *et al.* 2004[76], Li 2001 *et al.*[182]).

By using a new database for counting the itemset support, the *AprioriTID* algorithm, developed by Agrawal and Srikant 1994 [3], extends *Apriori* by eliminating multiple passes over of the database D. The new database has the form $< TID, C_k^i >$, where TID is the identifier of an itemset, and C_k^i represents the subsets of the itemset TID of length k. Thus, transactions are represented by the k-itemsets that describe them. For example, if D = $(bread, milk, apples)$, then equivalently C_1^i = ({bread}, {milk}, {apples}, {pears}), however, C_2^i contains all potential 2-itemsets within D, hence C_2^i= $(\{bread, milk\}, \{bread, apples\}, \{bread, pears\}, \{milk, apples\}, \{milk, pears\}, \{apples, pears\})$. It is apparent from this example that this new database may be larger than D. Thus, if C_k^i fits in memory *AprioriTID* is faster than *Apriori*, but when C_k^i is too big it cannot fit in the memory, and the computation time becomes much longer than *Apriori*. A new algorithm called *AprioriHybrid* was proposed to fulfil the *AprioriTID* algorithm and the *Apriori* algorithm drawbacks.

As its name denotes, *Partition* algorithm, developed by Savasere *et al.* 1995 [243] is based on the idea of partitioning the database in several parts which may be fitted into the memory. To be frequent in the complete data, an itemset must be at least frequent in one part. The *Partition* algorithm computes all itemsets that are frequent locally within a part. Therefore the *Partition* algorithm analyses each part like the *Apriori* algorithm, except that the itemset supports are not computed by counting

the occurrences. Indeed, the part is written in a better format suited to memory processing which reverses individuals and variables – instead of describing each individual with the items, the algorithm describes each item with the list of individuals which make it, called the*id-list* (every individual is designed by an id). Each item also has its *id-list*, and the number of occurrences of an itemset is given by the number of inputs in it's *id-list*. So to determine the support of an itemset I in a part, it suffices to calculate the intersection between the id-lists of two of its sub-itemset A and B such that $I = A$ and B. After analysing all parts, *Partition* checks for each itemset found frequently on a one part if it is often frequent in the complete data. Ultimately, the algorithm makes only two passes over the full data: the first to partition the data and the second to verify the frequent itemsets.

The *Dynamic Itemset Couting (DIC)* algorithm, introduced by Brin *et al.* 1997 [49], is a relaxed version of *Apriori*. The *DIC* algorithm reduces the number of passes over the database by introducing a new interesting idea: *possible frequent itemsets* – $(k + 1)$ candidates are computed from the k pass. When a k-itemset is considered frequent, all the $(k+1)$-itemset candidates that the latter can produce are generated. The basic principle is: during a pass over the data, if the occurrences of an itemset is already sufficiently high so that we know that it is common, then the itemset can already be used as a candidate for the next level. Therefore, counting occurrences of k-itemset candidates will be started as soon as the counters of all its sub-itemsets of length k-1 are high enough toward the support threshold. If the counting of the new candidate begins on the x-th individual, it will stop on (x-1)-th individual in the following passes. Finally, *DIC* makes fewer passes over the data then *Apriori*.

A comparison of the main association rule mining algorithms has been made by Hipp *et al.* 2000 [58], and by Goethals and Zaki 2003 [122] on various real and synthetic data sets. They show that the algorithm performance may vary according to the studied data. There is no one better algorithm than an other.

1.4.2 Constraint-based Association Rule Mining

At the same time when exhaustive algorithms were being developed, constraint-based algorithms which were introduced by Fu and Hah 1995[111]. Constraint-based algorithms extract specific association rules with additional constraints in conjunction with support and confidence thresholds. Nevertheless, even if these two constraints are basic, it is quite difficult to find the right values which produce interesting rules. Using wrong thresholds could have two consequences: firstly, the algorithm may miss some interesting rules and, secondly, it could generate trivial ones. Furthermore, users could have difficulties to understand the meaning of data-oriented constraints and to set minimal thresholds for these constraints. Constraint-based mining provides the user with the possibility to impose a set of constraints over the content of the discovered rules. Those constraints can significantly reduce the exploration space while improving the quality or interest of the results.

1.4.2.1 Constraints

Constraint-based algorithms use constraints to reduce the search space in the frequent itemset generation step (the association rule generating step is identical to that of exhaustive algorithms). The most common constraint is the support minimum threshold. If a constraint is reflexive, its inclusion in the mining process can provide significant reduction of the exploration space due to the definition of a boundary within the search space lattice, above or below which exploration is not required. Generally, constraints ($\mathcal{C}$) are provided by means of different formalisms: user knowledge constraints, data constraints, dimensional constraints, interestingness constraints, and rule constraints.

The role of constraints is very well-defined: they generate only association rules that are interesting to users (Zhao and Bhowmick 2003 [310], Hipp and Güntzer 2002 [145]). The technique is quite trivial: the rules space is reduced whereby remaining rules satisfy the constraints. If we take the example of the shopping basket, the constraints can be:

1. association rules must end with a product whose price is lowest *minprice*;
2. the total price of products in the rule antecedent must be less then the total price of the products in the rule consequent;
3. the rules must include only textile products.

These examples illustrate two important issues. On the one hand, the constraints can focus on a numerical characteristic of the items (product prices in the examples 1 and 2). On the other hand, the constraints can be expressed using a taxonomy of concepts describing the items (in example 3, a taxonomy that distinguishes textile products).

The two main categories of constraints that have been studied are anti-monotonic constraints and monotonic constraints:

- monotonic constraints are constraints that when valid for an itemset are inevitably valid for any of its super-itemsets. For example, consider a rule constraint ($sum(S) \geq 50\$$). If an itemset (S) satisfies the constraint that is the sum of the prices in the set is greater then 50$, further additional items to S will increase the cost and will always satisfy the constraint.
- anti-monotonic constraints are constraints that when invalid for an itemset, are also invalid for any of its super-itemsets. For example, consider a rule constraint ($sum(S) \leq 50\$$), if an itemset (S) does not satisfy the constraint (the price summation of the itemset is more than 50$), then this itemset can

be pruned from the search space since adding more items into the set will make it more expensive and thus will never satisfy the constraint.

Therefore, the monotonic constraint is the negation of the anti-monotonic constraint and vice-versa. Thus, a constraint-based algorithm is optimised to one constrain category among others.

In relation with itemset identification, a reflexive constraint is one that never decreases (monotonic constraint) or increases (anti-monotonic constraint) as the number of items within an itemset increases. Reflexive constraint inclusion can, therefore, reduce search space by effectively eliminating all super-itemsets of an invalid itemset. For example, the support threshold constraint is an anti-monotonic constraint. Given an invalid itemset whose supportitemset ¡ *minsup*, all super-sets of this item can be eliminated. The support constraint was first introduced within the *Apriori* algorithm (Agrawal and Srikant 1994 [3]). Examples of monotonic and anti-monotonic constraints are shown in Table 1.4.2.1.

Monotonic constraints	Anti-monotonic constraints
$S \supseteq I$	$S \subseteq I$
$min(S) \leq V$	$min(S) \geq V$
$max(S) \leq V$	$max(S) \leq V$
$sum(S) \leq V$	$sum(S) \leq V$
$length(S) \geq V$	$length(S) \leq V$

TABLE 1.5: Examples of monotonic and anti-monotonic constrains on an itemset S. I is a set of items, V is a numeric value

1.4.2.2 Algorithms

Most of the constraint-based algorithms are generalisations of the exhaustive *Apriori* algorithm (Agrawal and Srikant 1994 [3]). As *Apriori* used an anti-monotonic constraint (support), all constraint-based algorithms descended from *Apriori* can exploit these constraints effectively.

An anti-monotonic constraint can be used before the pass over the data (after line 4 of the algorithm Table 1.4.1.1), because it needs data reading (like the support minimum threshold constraint tested in line 9 of the algorithm 1.4.1.1). However, a monotonic constraint can be used after itemset candidate generation (between line 3 and line 4 in the algorithm Table 1.4.1.1), because it does not need data reading. Monotonic constraints are able to generate itemsets without generating candidate itemsets which make them more difficult to use.

The problem with monotonic constraints as presented by Jeudy and Boulicaut 2002 [160] is that monotonic constraints can deteriorate the anti-monotonic constraints by pruning performance (see example below). In this situation, the introduction of monotonic constraints in the process of frequent itemset generation increases the size of the research space instead of decreasing it. An other approach that can be more efficient is to use constraints in the post-processing of the association rule stage – after generating the frequent itemsets, to filter the obtained results.

Example 1.4.3 Let us consider the data described by a set of items $I = (Milk, Bread, Eggs)$. The anti-monotonic constraint C_1 is the minimum support threshold, and the monotonic constraint C_2 is a constraint which requires that the item $Milk$ must be in the itemset.

At the end of the algorithm's first iteration (level $k = 2$). The sets of itemsets that verify the two constraints are $F_2 = (\{Milk,Bread\}, \{Milk,Eggs\})$. The itemset $\{Bread,Eggs\}$ is does not in F_2 – it not verify the constraint C_2 and, thus, it is not generated as a candidate for the next level. In the next iteration (level $k = 3$), the set of candidate itemsets $F_3 = (\{Milk,Eggs,Bread\})$ is generated, then it should be pruned because it does not verify the constraint C_1. Suppose that $\{Bread, Eggs\}$ does not verify the constraint C_1. This information is unknown by the algorithm since this itemset has not been generated as a candidate from the lower level. Therefore it is impossible to predict that $\{Milk, Bread, Eggs\}$ does not verify C_1. When in doubt, $\{Milk, Bread, Eggs\}$ is kept in F_3. The problem is that some itemsets can be discarded after using the monotonic constraint C_2 whereas the anti-monotonic constraint C_1 could prune large parts of the search space.

These general principles for constraint use has been exploited by different researchers. The *CAP* algorithm proposed by Ng *et al.* 1998 [207] investigated applying item constraints (monotonic constraints) to generate frequent itemsets. They restricted the items or the combinations of items that are allowed to participate in the mining process. Earlier, in 1992, Smyth and Goodman 1992 [254] described a constraint-based rule miner integrating an interestingness constraint described by the dimension of rules, thus, long rules are considered less interesting. The *FP-growth* algorithm has also been adapted in constraint-based algorithm. *FIC* (Pei and Han 2000 [218]) and *FPS* (Leung *et al.* 2002 [180]) (extensions of FP-growth algorithm) were proposed to exploit constraints for mining constrained frequent sets. Jeudy and Boulicaut 2002 [160] proposed an extension of the *Apriori* algorithm that uses a conjunction of monotonic and anti-monotonic constraints. Finally, despite these various publications, finding the best way to exploit the constraint combinations of anti-monotonic constraints and monotonic constraints whatever the data remains an open problem.

1.4.3 Which approach to choose ?

If the user wants to exploit constraints in order to target association rule extraction results that interest him/her, then he/she must choose between two alternatives:

- run a constraint-based algorithm;
- run an exhaustive algorithm, then apply constraints in the post-processing step to filter the algorithm results.

The first solution presents two advantages. The first one is that using constraints allows the research space to be reduce. This solution avoids consuming resources and time for association rules that do not interest the user. The second one is that in some cases constraint-based solutions are the only solutions proposed to process the data with the same minimum support threshold as exhaustive algorithms. The combinatorial explosion of frequent itemset numbers makes it impossible for exhaustive algorithm execution. Then – since pruning do not only use the minimum support threshold constraint, it is possible for constraint-based algorithms to use even lower minimum support thresholds. For instance, very specific association rules can be generated by a constraint-based algorithm which would not be discovered by an exhaustive algorithm like *Apriori* because of the combinatorial explosion. The possibility of applying very specific rules often provides unusual and unknown knowledge for the user, which make it very interesting (Freitas 1998 [109]).

The second solution has also his own advantages. The first advantage as we saw is that constraint-based algorithms can not exploit optimally anti-monotonic constraints in all situations. Sometimes, applying constraints in the post-processing stage can be faster than generating results with a constraint-based algorithm. The second advantage takes its strength from the iterative and interactive nature of the data mining process itself. In the data mining process, the user can multiply successive association rule extractions. Every new extraction query depends on the antecedent query results. If an exhaustive algorithm can process all frequent itemsets in an acceptable response time than all frequent itemsets can be available for any query request by the user. It can be sufficient to generate association rules – if the minimum support threshold is not decreased, from the frequent itemset already calculated. Consequently, if the frequent itemset generation stage is processed completely, the user doesn't care about the execution time, and does not change the minimum support threshold, this solution can promote response time (Geothales and Bussche 1999 [121]).

1.5 Problematic of Association Rules and Solutions

The main disadvantage of the rule extraction process is that the volume of generated rules often greatly exceeds the size of the underlying database. Typically only a small

fraction of that large volume of rules is of any interest to the user who is very often overwhelmed with the massive amount of rules. Cognitive processing of thousands of rules takes much more time then generating them even by a less efficient tool. Imielinski 1998 [152] believes that the main challenge facing association rule mining is what to do with the rules after having generated them.

Association rule generation methods have to face two types of problem:

- rule quantity: the huge number of mined association rules makes manual inspection practically infeasible. It also increases the difficulty in interpreting the results and obtaining relevant knowledge.
- rule quality: an equally or possibly more important issue concerns the quality of the extracted rules. Rules such as "$age = 10 \rightarrow unemployed$", while being statistically valid in a database are obvious since they are trivial knowledge. For instance, Major and Mangano 1995 [189] mined 529 rules from a hurricane database of which only 19 were found to be actually novel, useful and relevant.

There has been various research aimed at the attenuation of both problems (rule quality and rule quantity). Proposed solutions for the rule quality problem rely on the specification of interestingness measures to represent the novelty, utility or significance of rules. By ordering the discovered rules according to their degree of interestingness, one can ensure that only good quality rules are presented first to the analyst. For the rule quantity problem, various strategies have been proposed, among them being redundancy reduction and post-processing techniques, these being the two most widely used.

1.5.1 Interestingness Measures

In association rule mining the user often needs to evaluate an overwhelming number of rules. Interestingness measures are very useful for filtering and ranking the rules presented to the user. In the original formulation of association rule discovery problem, support and confidence are two of the interestingness measures proposed.

Support is necessary because it represents the statistical significance of a pattern. From the marketing perspective, itemset support in a market database justifies the feasibility of promoting the items together. Support is also good for pruning the search space since it possesses a nice downward closure property (anti-monotonic constraint). Tan and Kumar 2000 [209] examined various objective interestingness measures and demonstrated that support is a good measure because it represents how statistically significant a pattern is. Support-based pruning is effective because is an anti-monotonic function, and it allows us to prune mostly uncorrelated or negatively correlated patterns. Beyond that, it may not serve as a reliable interestingness measure. For example, rules with high support often correspond to obvious knowledge about the domain and consequently, may be not interesting to the analyst simply

because it does no reveal any surprising information. In theory, confidence measures the conditional probability of events associated with a particular rule. Unfortunately, confidence can be misleading in many practical situations, as shown by Brin *et al.* 1997 [48].

Except for these two measures, many other have been proposed in the literature. Geng and Hamilton 2006 [117] have classified interestingness measures into three categories : objective, subjective, and semantics-based measures.

- Objective Measures Based on Probability: objective interestingness measures are based on probability theory, statistics, and information theory. Therefore, they have strict principles and foundations and their properties can be formally analysed and compared. Objective Measures take into account neither the context of the domain of application nor the goals and background knowledge of the user. They evaluate the generality and reliability of association rules. Hilderman and Hamilton 1999 [144] surveyed 70 interestingness measures (mostly objective) that have been successfully employed in data mining application.
- Subjective Measures: in application where the user has background knowledge, rules ranked highly by objective measures may not be interesting. A subjective interestingness measure takes into account both the data and the user's knowledge. Such a measure is appropriate when:
  - the background knowledge of users varies;
  - the interests of the users varies;
  - the background knowledge of users evolves.

  Unlike objective measures, subjective measures may not be represented by simple mathematical formulae because the user's knowledge may be representable in various forms. Instead, they are usually incorporated into the mining process. Subjective measures can be classify in:
  - Surprisingness
  - Novelty
- Semantic Measures: semantic measures consider the semantics and explanations of the association rules. Geng and Hamilton 2006 [117] consider that semantic measures are based on:
  - Utility
  - Actionability

Later in 2009, Blanchard *et al.* [30] proposed a Semantics-based classification of objective interestingness measures. They proposed to classify interestingness measures according to the *subject*, the *scope* and the *nature* of the measure. This means that an interestingness measure is evaluated according to the notion that it measures, the elements concerned by its results, and the type of the measure: descriptive or statistical.

During the data mining process, interesting measures can be used in two ways. Firstly, objective measures can be used to prune uninteresting patterns during the mining process to reduce the search space. However, this may eliminate interesting rules. For instance, the support threshold can be used to filter out patterns with low support during the mining process. The direct consequence is that some items are not taken into account if they are not frequent in the database although they could be interesting to the user. By eliminating these items the user can waste valuable information. Secondly, measures can be used during post-processing to select interesting patterns. For instance, we can rank rules by various interestingness measures and keep only rules with the higher ranking. The problem is that rules that have the highest rank are not necessarily the good ones. Much research work (Ohsaki *et al.* 2004 [211], Tan *et al.* 2002 [268]) has compared the ranking of rules by human experts to the ranking of rules by various interestingness measures, and suggested choosing the measure that produces the ranking that most resembles the ranking of experts. This problem represents a major gap between the results of data mining and the use of the mining results. Interestingness measures can play an important role in the identification of novel, relevant, implicit and understandable patterns from the multitude of mined rules. They help in automating most of the post-processing work in data mining. But it should be the analyst who validates rules and designates the more interesting ones.

1.5.2 Redundancy Rule Reduction

This second section of solutions for the volume of generated rules is dedicated to redundant rule reduction techniques. Classical algorithms propose good methods for association rule extraction, but the number of rules is too large and too many rules are redundant.

Redundancy reduction refers to a set of techniques aimed at pruning out items or rules which do not reveal interesting knowledge. Pruning techniques can be used to dynamically reduce the dataset during processing. By discarding unnecessary items, items pruning generates dataset reduction (D) and further reduces processing time. Pruning can deal with the quantity problem, if a set of rules means the same thing or refers to the same feature of the data, then the most general rule may be preserved.

Rule covers (Toivonen *et al.* 1995 [275]) is a method that retains a subset of the original set of rules. This subset refers to all transactions (in the database) that the original rule covered. Another strategy in association rule mining was presented by

Zaki 2000 [305] based on the concept of closed frequent itemsets that can drastically reduce the generated rule set. Therefore, the set of generated frequent itemsets is smaller than the set of all frequent itemsets, especially for dense databases. The closed frequent itemsets are used to generate a set of rules, from which all other association rules can be inferred. Thus, only a small set of rules can be presented to the user. These basic rules gives a user only an overview of the domain. Although the idea was proposed by Zaki 2000 [305], it is in 2004 that Zaki 2004 [304] developed a real algorithm.

Comparable with the algorithm proposed by Zaki 2000 [305], Pasquier *et al.* 2005 [217] introduced two condensed associations based on closed frequent itemsets to represent non-redundant association rules. These two representations called*Min-max Approximative Basis* and *Min-max Exact Basis* describe different possibilities to extract non-redundant rules using closed methods. Later, Xu and Li 2007 [298] improved the definitions suggested by Pasquier *et al.* 2005 proposing a more concise association rule basis called *Reliable exact basis*.

Li 2006 [181] proposed *optimal rule sets* defined with respect to an interestingness measure. An optimal rule does not have a more general rule with greater interestingness. Thus, the algorithms produced a set of rules contains all optimal rules.

A set of techniques for the reduction of redundant rules was developed and implemented by Ashrafi *et al.* 2005 [12]. The proposed techniques are based on generalisation/specification of antecedent/consequent of the rules and they are divided in methods for multi-antecedent rules and multi-consequent rules.

Several researches suggested that pruning redundant rules can seriously decrease the number of redundant rules is exponential (Zaki 2004 [304]). Redundancy reduction methods may not provide an overall picture if the number of the pruned rules is too large.

1.5.3 Interactive Rule Post-processing

Rule post-processing involves helping user to select rules which are relevant or interesting and, thus, reduce the cognitive load of experts. But, since it is always the user who decides what is interesting this means that post-processing of association rules a non trivial task (Roddick *et al.* 2001 [239], Freitas 1999 [107], Bayardo *et al.* 1999 [19], Dong and Li 1998 [92], Silberschatz and Tuzhilin 1995 [247], Silberschatz and Tuzhilin 1996 [249]).

Many authors have stressed that the data mining process is by nature highly iterative and interactive and requires user involvement (Fayyad 1997 [100]). In particular Brachman and al. 1996 [45] have pointed out that in order to efficiently assist the users in their search for interesting knowledge, the data mining process should be considered not from the point of view of the discovery algorithms but from that

of the users, as a human-centred decision support system. Thus, to find interesting rules it is more efficient to incorporate the user into the *Data Mining* process than just presenting filtering operators to help the user exploring the association rule extraction algorithm results. Yamamoto *et al.* 2009 [72] compared the performance of the user-driven approach I2S (Yamamoto *et al.* 2007 [299]) with a conventional rule extraction approach *Apriori* to illustrate the possibilities of the user-driven solution. In a case study with a domain specialist, the usefulness of incorporating the analyst into the data mining process was confirmed. We believe that a closer contact with the discovery process allows users to obtain interesting rules faster than just running an algorithm as a black box.

The main approaches for interactive association rule post-processing can be grouped into three classes:

1. organise an interactive exploration/extraction of association rules;
2. organise an interactive visual exploration of association rule mining results;
3. organise an interactive visual extraction of association rules.

1.5.3.1 Interactive Exploration and Extraction of Association Rules

Rule Browser

Interactive tools of the type *rule browser* have been developed to assist the user in interactive post-processing of association rules. Like the file browser, they are interactive interfaces that present information in textual form.

Sahar 1999 [242] proposed to the user to eliminate non interesting rules by removing not interesting items. If a user eliminates an item, all rules in the resulting list that contain this item are removed. The effectiveness of this approach lies in its ability to quickly eliminate large families of rules that are not interesting, while limiting user interactions to a few, simple classification questions. In 2000, Ma *et al.* 2000 [187] proposed a system where the user explores a summary of rules and he/she can select interesting rules from it. More recently, a feature-rich rules browser was presented by Fule and Roddick 2004 [115] (Figure 1.4). It allows the user to filter association rules by applying more or less general syntactic constraints as they may use a taxonomy of items. The tool also allows the user to choose what rule interestingness measure should be used for sorting and filtering rules. Furthermore, the user can save the rules that he/she deems interesting during the exploration.

Some association rule browsers exploit subjective interestingness measures and then can profit from the user knowledge. For example, Liu *et al.* 1999 [184] proposed a rule browser based on subjective interestingness measures. This tool exploits the user's knowledge of the data domain to present the rule. Firstly, the user expresses

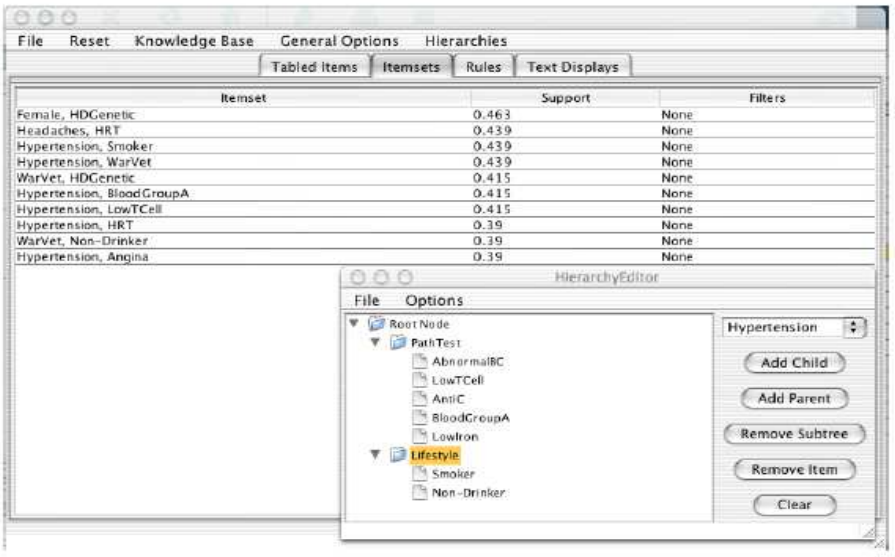


FIGURE 1.4: The RSetNav rules browser [115]

his knowledge under the form of relations by specifying a set of patterns according to his/her previous knowledge or intuitive feelings. Then, the tool matches and ranks the discovered rules in different categories according to whether or not they confirm the user's wishes. In general, the technique ranks the discovered patterns according to their conformity to the user's knowledge or their unexpectedness, or their actionability. The assumption of this technique is that some amount of domain knowledge and the user's interests are implicitly embedded in his/her specified pattern. With such a ranking system, the user can check the lists of rules to confirm his/her expectation, or to find those patterns that are against his/her intuitions, or to discover those rules that are actionable. This method can be used to confirm user knowledge, to find unexpected patterns or to discover actionable rules. The main limitation of the rules browser is their textual representations of the rules which does not suit the study of large amounts of rules. These tools also have the problem of implementing only a few interestingness measures (maximum of three).

Query Languages

The concept of *Inductive Databases*, was introduced by Imielinski and Mannila 1996 in their founding article [151]. The main idea was to extend the database management system in order to incorporate data mining methods. In other words, it is about developing a data mining query language, an SQL generalisation that allows data creation and manipulation and also knowledge discovery from databases (classify, clusters of association rules, etc.). For the user, he/she manipulates a database containing both data and knowledge, regardless of whether the information is actually stored in the database or dynamically generated from data.

The inductive database was an ambitious and difficult project. Despite research since 1996, there are still many challenges to meet. Query evaluation and optimisation are particularly difficult. (we saw in Section 1.4.3 that it is difficult to optimise the extraction of rules if we do not know the optimal constraint of thresholds in advance). Many query languages have been developed in the context of *Inductive Databases* to extract and manipulate association rules, such as DMQL (Hah *et al.* 1996 [133]), MSQL (Imielinski and Mannila 1996 [153]), MINERULE (Meo *et al.*1998 [193]), and XMINE (Braga *et al.* 2002 [46]). These query languages allow the user to control the extraction and/or post-processing of association rules. However, they can be impractical as regards the post-processing of association rules, in the same way as SQL used alone is not suitable for data analysis.

To remedy this, Bonchi *et al.* 2009 [34] presented ConQuestSt, a constraint-based query system for exploratory and interactive association rule discovery, where the interest of the rules is defined by means of user-defined constraints. The user can supervise the whole process by defining the parameters of the three tasks (pre-processing, data mining and post-processing) and evaluating the quality of each step and, if necessary, re-adjusting the parameters. The tool proposed by the authors offers many constraint-based association rules and offers the SPQL query language for pattern discovery as introduced in Bonchi *et al.* 2006 [33] and extended in Bonchi *et al.* 2009 [34]. Moreover, via the graphical interface of the ConQuestSt tool (Figure 1.5) the user can store interesting rules in the DBMS.

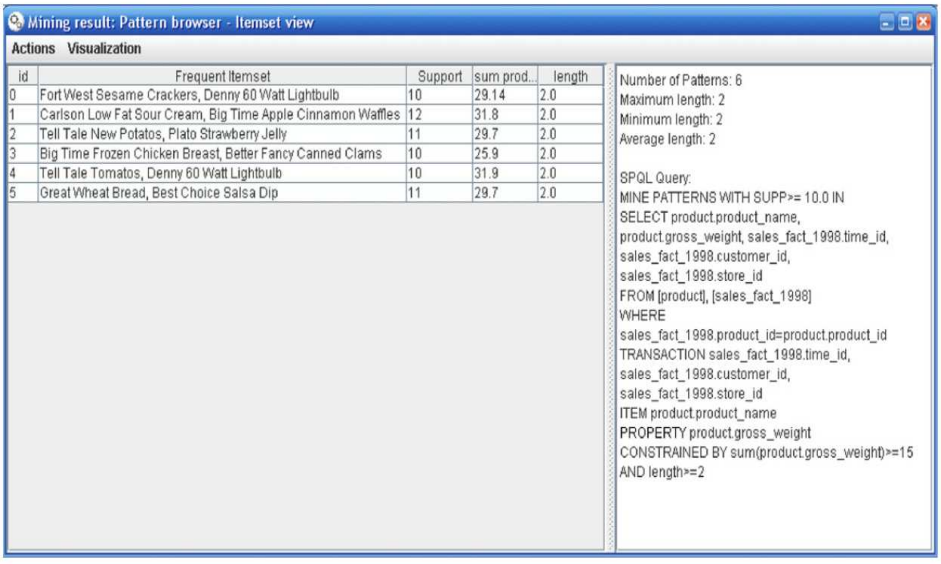


FIGURE 1.5: The ConQuestSt's pattern browser window [34]

1.5.3.2 Interactive Visual Exploration and extraction of Association Rules

In the post-processing of association rules, it is often through rule visualisation that post-processing is performed. Visualisation take advantage of the intuitive appeal of visual representations such as graphs, colour and charts to attract the user's attention. In addition, innovative ways of depicting the results of data mining in two or three dimensions help provide more realistic information than textual representation (for more details see Chapter 2 Section 2.4).

Visual representations of data mining results also allow easily interaction. Association rule representations easily allow navigation at various levels of details by interactively and iteratively changing the rule views. This mean that different scenarios can be analysed and compared. Groups of rules can be validated on the basis of their visual representations. This can reduce the cognitive load of the experts when dealing with many rules.

Visual representations can be used:

- in conjunction with data mining algorithms to facilitate and accelerate the knowledge analysis;
- as a method of data mining.

Visualisation of association rule mining results

Typical visual representations to display results of association rule mining are grid-like structures and bar charts. The grid view consists of 2D matrix of cells where each cell represents a rule. One matrix dimension represents rule antecedent and the others represent rule consequent. Each cell is filled with coloured bars indicating rule support and confidence values. However, this representation often suffers from occlusion. Besides, it is difficult to represent rule antecedent or consequent with more then one item. This restriction can be overcome by extending the matrix dimensions to represent combinations of items, but it becomes impracticable if there are too many different items.

Kopanakis and Theodoulidis 2003 [171] introduced a representation using bar charts coupled with a parallel coordinate visualisation (Inselberg and Dimsdale 1990 [155]) and a grid view. The bar charts (Figure 1.6(a)) map both interest measure values and rule contents. Each item value (discrete or continuous) from both the antecedent and the consequent is visually mapped as a bar chart. If the element is discrete, then a value is indicated in the bar; if it is continuous, an interval is indicated instead. This approach represents only one rule at a time, and a grid of representations (Figure 1.6(b)) is used to display multiple rules. A distance measure is employed to evaluate rule similarity in order to determine their placement on the grid. The similar rules are placed close to each other. The underlying rationale in the

parallel coordinate visualisation (Figure 1.6(c)) technique is to map data attributes (transaction items, in this case) to parallel axes, and map each datum instance (rule, in this case) as a polyline intersecting each axis at a point corresponding to its particular value. This approach offers a different perspective to visualise rules but remains limited to a relatively small set of rules.

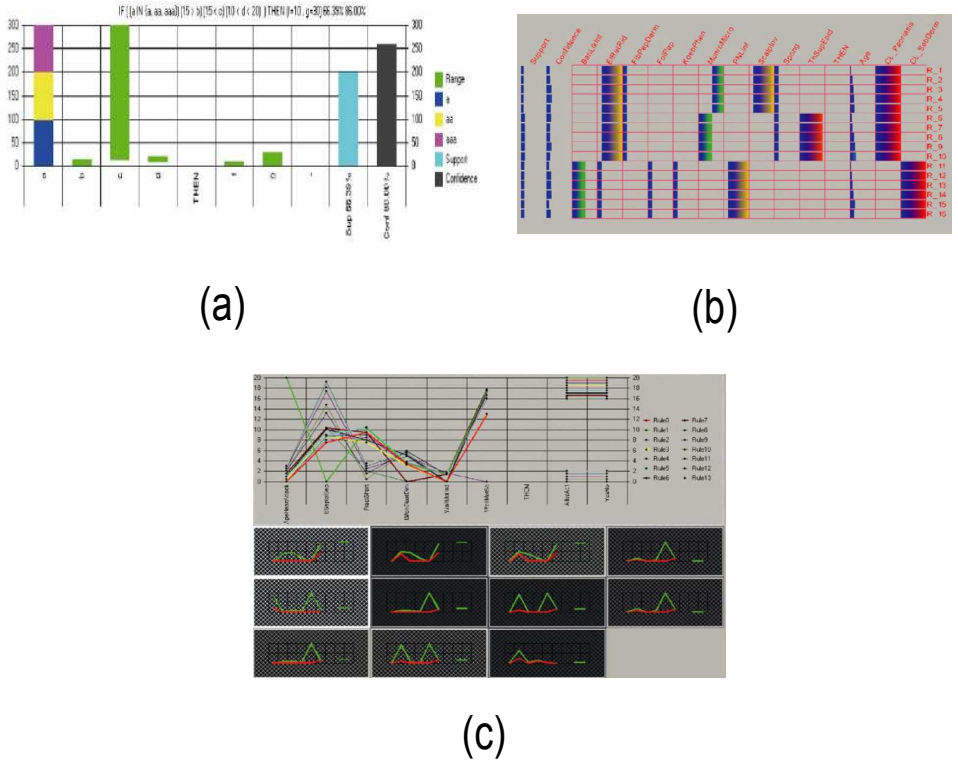


FIGURE 1.6: An association rule representation using bar chart for one rule visualisation (a), grid-like visualisation for multiple rules visualisation (b) and parallel-coordinate visualisation (c) [171].

Wong *et al.* 1999 [297] introduced a representation based on a two-dimensional matrix that maps rule-to-item rather then item-to-item relationships. Matrix rows represent items, whereas matrix columns represent rules. A block placed at the appropriate row-column intersections depicts a particular rule-to-item relationship. Blue blocks indicate rule antecedent and red ones indicate rule consequent. A 3D view of the matrix is displayed (Figure 1.7), with support and confidence values represented as a bar chart placed in the scene background. Chakravarthy and Zhang 2003 [64] proposed a two-dimensional matrix view similar to the previous one by Wong *et al.* 1999 [295] the major difference being that one of the matrix dimensions now represented rule consequent. In the rules representation proposed by Chakravarthy and Zhang 2003 [64], the rules are grouped according to the number of items at their antecedent, and bar charts represent their interest measures, namely, support and confidence.

Each bar represents a quantity of items at the antecedent, and is sub-divided into interest measure range value (e.g., 60-70% confidence, 80-90% confidence, etc). This representation gives a broader view of the rules set, and the user may obtain a closer look by clicking on a bar, which causes a two-dimensional matrix view. Couturier *et al.* 2007 [79] combined 2D matrix for visual representations of association rules with graphical fisheye view. Their goal is to give users a details-on-demand view while preserving an overview of the overall context. One visualisation area shows rules in a 2D matrix, with antecedent and consequent mapped to each dimension and values for interestingness measures (support and confidence) displayed as colours within the corresponding cell, whose visual area is equally split among measures. The same authors Couturier *et al.* 2007 [80] proposed an approach for exploring association rules based on a visual representation of cluster of rules, obtained by applying the k-means algorithm over the rule characteristics. Each cluster is represented in a grid-cell, and colours map interest measure values.

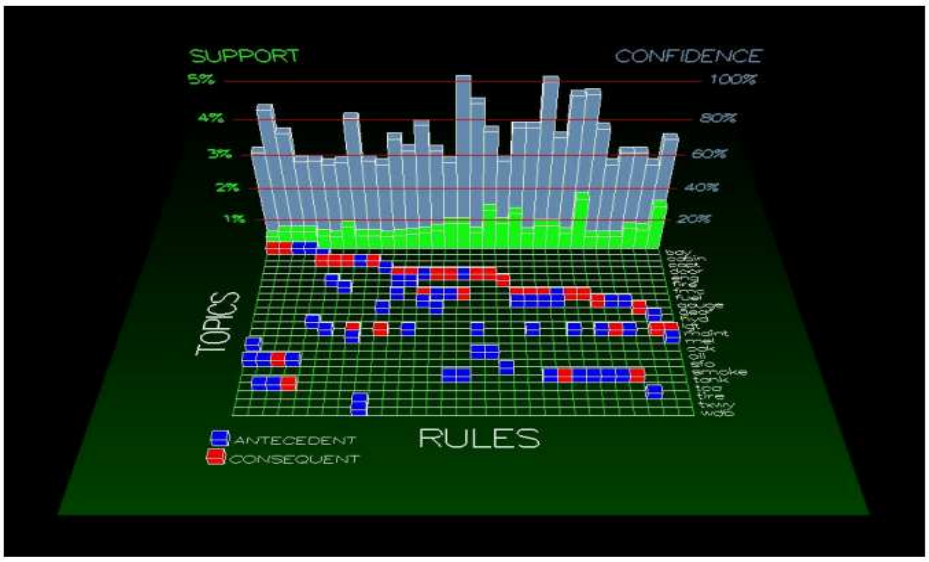


FIGURE 1.7: Visualisation of item associations [297].

Hofmann *et al.* 2000 [146] introduced a representation using *Mosaic plots* (Figure 1.8). They display a contingency table to represent an association rule by mapping the rule support and confidence as areas proportional to their values. The representation provides deeper understanding of the correlation between the antecedent and the consequent of a rule. This approach considers only one itemsets at a time, and interpreting the visualisation becomes difficult when three or more items appear either at the antecedent or the consequent.

Unwin *et al.* 2001 [280] proposed the *TwoKey plot* a scatter plot visualisation (Figure 1.9) of the corpus of association rules. The two most common interest measures,

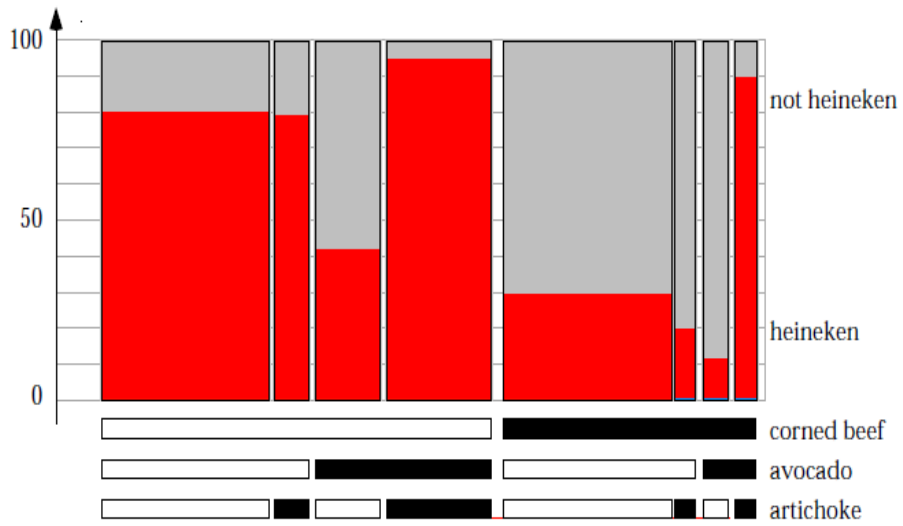


FIGURE 1.8: Association rules representation with Mosaic Plots [146].

namely support and confidence, are mapped respectively to the X and Y dimension of the plot. Therefore, a graphical marker representing a rule is positioned in the plot according to its support and confidence values, while the marker's colour maps the number of item's in the rule. This representation provides an overview of the whole rule set, and users can employ filtering operators to direct rule exploration, e.g., obtaining rules related to a rule query, such as its more specific rules and more general rules.

Ong *et al.* 2002 [148] adopted a grid-like visual representation for association rules with columns ranked by confidence values and rows ranked by support. The rules are grouped in cells, according to their support and confidence, and represented as squares whose locations are determined by their support and confidence values. Filters based on support, confidence or item selection assist user navigation through the rule set. The authors pointed out that their method shows rules of any size and with no limitations of rules number. However, only the interest measure value is visually represented (the itemsets are not represented). The same authors proposed a tree-like view of rules. The rules are grouped according to their antecedent and consequent. Antecedent are firstly shown as nodes, according to the number of items. Then, for each antecedent node, a rule containing the consequent is shown as sub-nodes. The Tree view represent the main contribution of Ong *et al.* 2002 [148] compared to the *TwoKey plot* proposed by Unwin *et al.* 2001 [280]. A grid-like rule visualisation (Figure 1.10) has been also employed later by Techapichetvanich and Datta 2005 [271], where two groups of lines represent items belonging to the antecedent and the consequent. Each rule is visually represented as a line that connects the item lines. If an item belong to the rule, the intersection of the rule line and the item line is

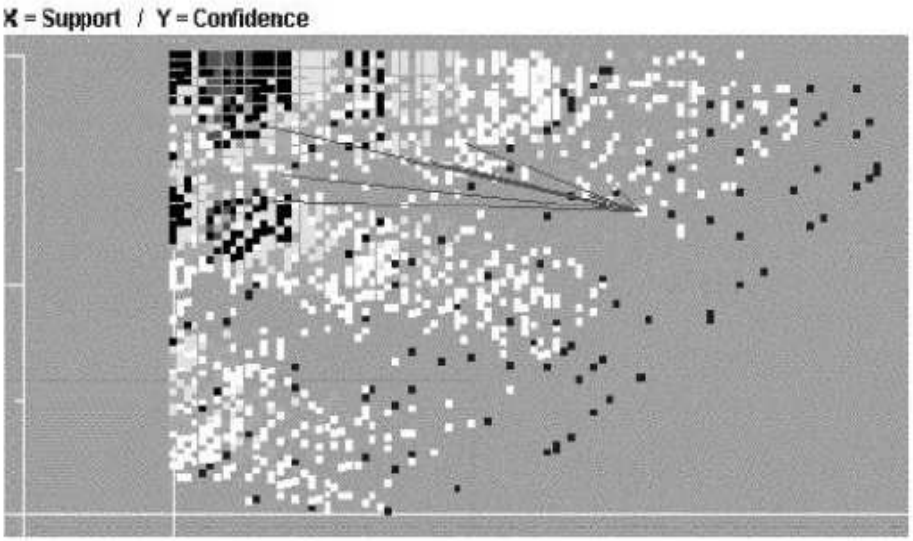


FIGURE 1.9: A scatter plot of 5807 rules with *TwoKey plot* [280].

marked by a dot. Either the support or the confidence value is divided into intervals and mapped to the colour of the line representing the rule.

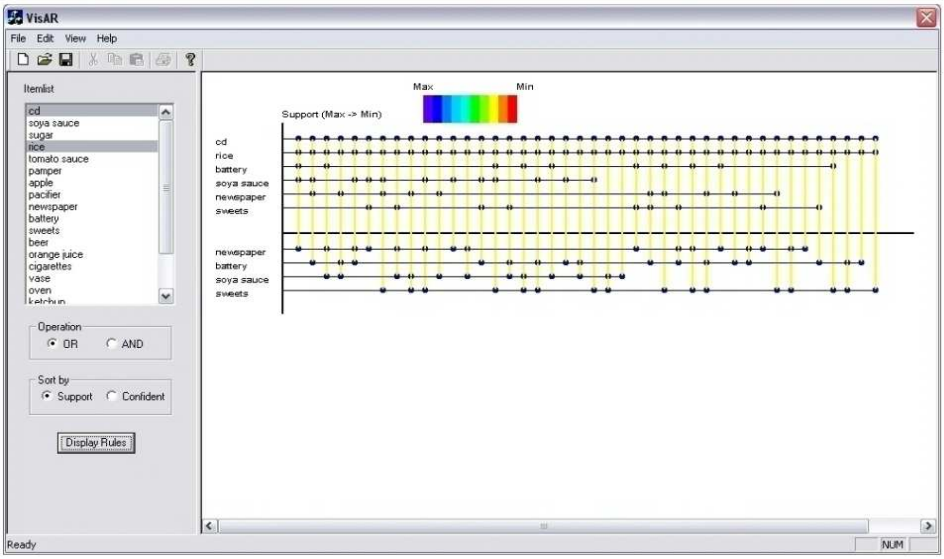


FIGURE 1.10: A grid-based visualisation of association rules [271].

Yang 2003 [300] employed parallel coordinate visualisation to display association rules (Figure 1.11). Rules are represented similarly, by two lines, one for the antecedent and another for the consequent. An arrow connects the lines, and colour

may map interestingness measure values. Support of a rule is represented by the line width. Confidence is represented by the colour.

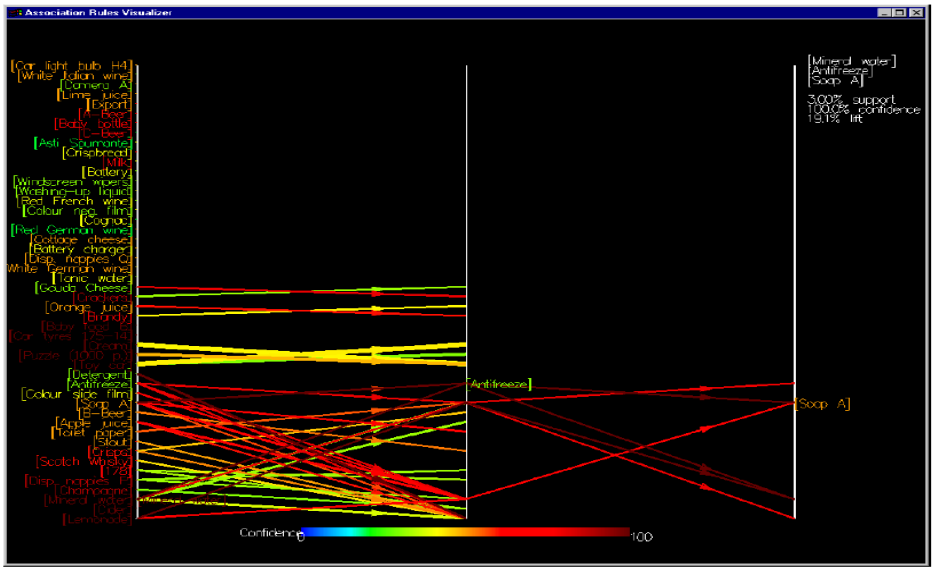


FIGURE 1.11: A parallel coordinates visualisation of association rules [300].

In the graph-based representation of association rules by Bruzzese and Buono 2004 [50] the colour of a node represents the antecedent (red) or the consequent (green) of a rule, while edges connecting the nodes depict the association. Confidence is mapped to the edge length, with colour mapping the support, light and dark blue referring to low and high support, respectively. A user may further explore subsets of rules using a parallel-coordinate representation. Unlike the previous visualisation approach, this representation allows the visualisation of a large set of rules while displaying the items making up the rules. Otherwise, it offers few interaction operators. Ertek and Demiriz 2006 [99] proposed to display association rules using a graph-based technique (Figure 1.12). Items and rules are mapped as nodes, and item-rule connections map their relations, with directed lines to indicate the direction of the implication. Rule node colour maps the level of rule confidence. The size of the nodes (area) show the support interestingness measure. The approach proposed by Yang 2003 [300] and Ertek and Demiriz [99] generate the results and then visualise both intermediate result–frequent itemset–and final results–association rules. However, they do not allow user interference during the analytical processing.

A few approaches provide visual representations of individual rules (Hofmann 2000 [146], Kopanakis and Theodoulidis 2003 [171]). They used reduced screen space, which make impossible to avoid a point of saturation when multiple rules are displayed and from where no rules can be further represented on the screen. This limitation are faced by all representations based on 2D matrix and on grids. On the other hand,

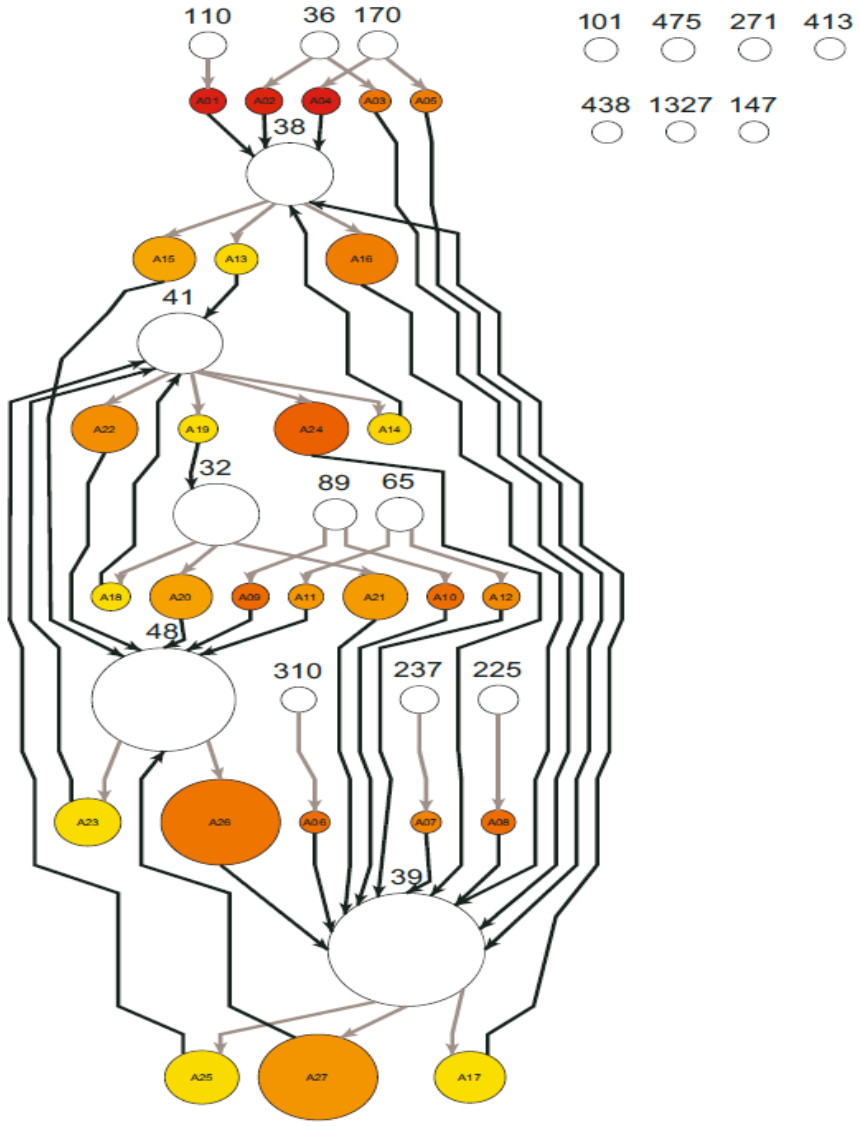


FIGURE 1.12: A graph-based visualisation of 27 association rules [99].

most Graph-based and projection-based approaches are less sensitive to the saturation problem, but suffer from object occlusion, i.e., graphical markers overlap, and proper interaction mechanisms are required to handle the problem.

Visualisation during association rules mining

Although approaches aimed at visualising the results of association rules mining help users to find interesting rules within the typically huge amount of rules, they do

not allow active user participation in the process, for instance by monitoring the association rule extraction algorithm. Active participation allows users to input previous knowledge into the data mining process itself, which can be done both by driving the generation of frequent itemsets, and the extraction of association rules. Some approaches that promote user introduction in the discovery process will be discussed.

The idea was first proposed by Klemettinen *et al.* 1994 [170] for analysis of failures in telecommunication networks. By using a browser, the user tries to reach interesting rules by revising interestingness measures thresholds and applying syntactic constraint (the user can choose the items that should appear or not appear in the rules) to reduce the number of rules. Then, bar charts are used to map interest measure values of the association rules. Bar height maps the interest measures values – support, confidence and their product (named commonness) so that high values catch the user's attention during rule exploration. The authors also present a graph-based visualisation of rules (Figure 1.13) where items are presented as nodes and associations as directed arrows. Arrow thickness represents rule confidence or, alternatively, support, and colour can map additional information. Nonetheless, the graph suffers from a cluttering problem when displaying too many rules. A solution to avoid this problem is to include merging multiple nodes from a cluster into a single node, or displaying nodes for the antecedent and the consequent in separate regions of the screen.

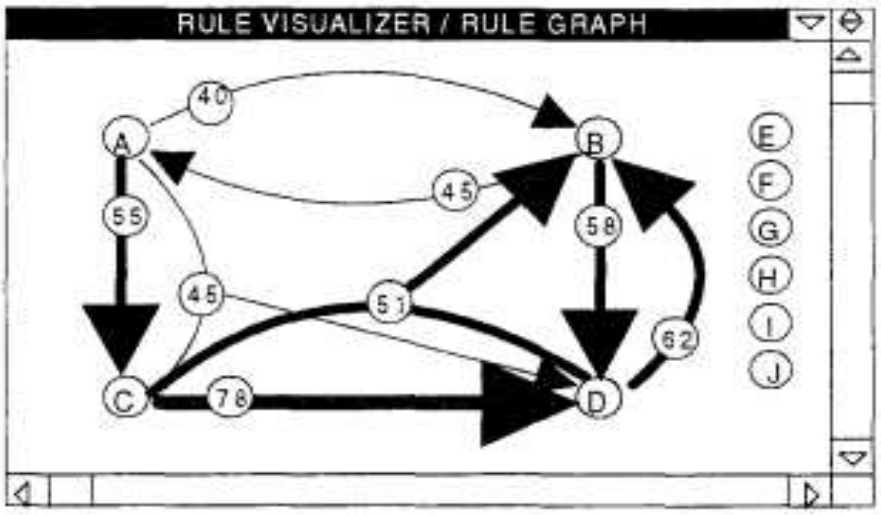


FIGURE 1.13: Rule visualisation / rule graph ([170]).

A step in this direction was taken by Liu and Salvendy 2006 [185], who allowed the exploration of frequent itemsets at each step of the *Apriori*, but the user cannot fully drive the process, for two main reasons. Firstly, rules are automatically generated for all the obtained frequent itemsets and it is not possible to focus on specific

groups of itemsets at this moment. Secondly, the user does not have full control over algorithm execution, since he/she cannot change the algorithm parameters during the process. In an interactive approach for association rule exploration, it is important for a user to be able to reverse the last action if he/she realises that it is not suitable. To face such a problem, it is desirable that the user have the possibility to backtrack (or forward) one step to change parameters or to restart the process with different parameters.

Kuntz *et al.* 2000 [174] introduced a new approach inspired by experimental work on behaviour during a discovery stage. The basic strategy of this approach is to start from the frequent items, similar to the *Appriori* algorithm. The user may then select items of interest and obtain rules involving these and other items (Figure 1.14). The rule extraction is dynamic: at each step, the user can focus on a subset of potentially interesting items and launch an algorithm for extracting the relevant associated rules according to statistical measures. This navigation operation is called forward chaining, and is graphically represented by graph-based visualisation (backward chaining is also supported). Besides being user-directed this strategy avoids generating unwanted rules. Blanchard *et al.* 2003 [28] proposed a user-centred rule exploration approach, which adopts a visual metaphor of two arenas to place interesting rules. The arena holds the generalised and specialised rules separately. Each rule is represented by a sphere, whose radius maps its support, and by a cone, whose base width maps its confidence. Additionally, the colours of the sphere and cone redundantly represent a weighted average of the measures, with the position of the rule at the arena represents the implication intensity. This work was extended later (Blanchard *et al.* 2007 [31]) with two complementary visualisations: the first is the 3D visual interface (see Chapter 3) and the other is the neighbourhood relationship between rules, some of them from the already mentioned work by Kuntz *et al.* 2000 [174]. Based on the neighbourhood relations, the authors proposed rules that are closer to a selected rule according to a neighbourhood relation. The available relations are: same antecedent, forward chaining, antecedent generalisation (which is opposite to forward chaining),same items, agreement specialisation, exception specialisation, generalisation and same consequent.

Yamamoto *et al.* 2007 [299] proposed a System called I2E that relies on an the interactive execution of *Apriori* aided by a visual representation of the space of itemsets. At each execution step the system identifies the frequent k-itemsets. Then, the users visualise and explore the proposed frequent itemsets via a graph-based visualisation of the itemsets extracted in the current step(k) . The itemsets that are similar in content are grouped together in the graph visualisation. After generating the frequent itemsets, the users can drive the rule extraction by changing the minimum support value and by filtering frequent itemsets. This approach supports a user-driven selective rule extraction that produces fewer rules, but still includes rules representative of all different groups of similar identified itemsets. After rule extraction, users can explore the complete rules space via a visual interface that allows pairwise rule comparisons.

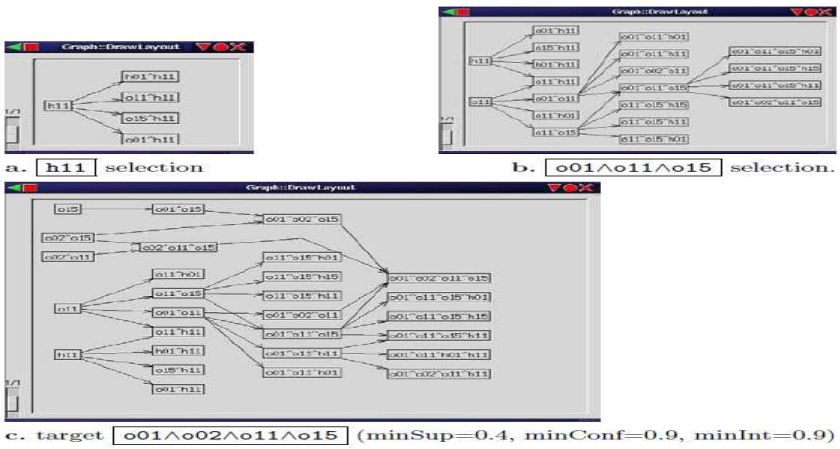


FIGURE 1.14: Discovering rules from the selected frequent items [174].

All these proposed approaches suffer from the same limitation. The user cannot record interesting association rules identified during the exploration and then he can't compare the interesting association rules. A typical approach to find interesting association rules needs its extraction from many subsets of data. Most approaches allow the visualisation of only one subset of association rules each time. Furthermore, the user cannot compare association rules extracted from different subsets of data. The user can forget interesting association rule discoveries if he cannot save and redisplay them to compare them to other association rules. On the other hand, the I2E system proposed by Yamamoto *et al.* 2007 [299] allows only pairwise comparison of rules using relations. This comparison allow only the comparison between two rules that have same syntactic relations– same antecedent, same consequent, rule generated from a subset of the generating itemset, rule generated from a subset of the generating itemsets and rules generated from the same itemset. Therefore, the user can't compare more then 2 rules and he can't compare two association rules which aren't syntactic link.

1.6 Conclusion

In this chapter, we have described processes of Knowledge Discovery in Databases and, more particularly, we have focused on the association rule mining techniques. We have presented the definitions, the notations, the extraction algorithms, and the most important solutions to answer drawbacks that limit the association rule mining technique.

The algorithms developed for association rule extraction fall into two categories: exhaustive algorithms and constraint-based algorithms. In absolute terms, neither approach clearly outperforms the other. The time and memory performance depend

on the extraction parameters and data studied. In practice, an exhaustive algorithm can be a suitable solution for hollow data sets or for users who can tolerate a long processing time but it still produces a huge number of rules. On the other hand, a constraint-based algorithm can produce a limited number of rules but requires that the user use the correct values of the constraints.

Different methods have been proposed to address these two problems:

- evaluate, order and filter rules with interestingness measures other than support confidence;
- reduce the number of proposed rules using redundancy reduction methods;
- organise an interactive exploration (and extraction) of rules.

Nevertheless, the two first approaches have their own limits. Firstly, the interest and the utility of a rule is decided by the user whereas the interestingness measures are generally based on database information. Redundancy reduction methods allow a reduction of the number of rules from millions to thousands, but even so, a user is not able to manually exploit thousands of rules. In conclusion, the *useful* characteristic of patterns suggested by Fayyad *et al.* cannot be met in this case.

Thus, the last technique allows the introduction of the user in the data mining process to extract reduced sets of rules which are useful and interesting for him. The most efficient approach for interaction exploration and extraction are based on visual interfaces. Moreover, on the one hand, the visualisations adopted in most reviewed approaches demand considerable screen space, which is quite limited in 2D interfaces. Given the large amount of visual objects it is impossible to show clearly all of the rules simultaneously. In addition, visualisation in 2D suffers from a lack of agreeable semantics for the position and orientation of objects when placed on a 2D grid, which can cause biased visual feedback.

On the other hand, a very limited number of approaches support a fully user-driven rule extraction process and no one provides the user with the possibility of saving and comparing the discovered interesting rules. Thus, new representation/interaction techniques are needed to extract reduced sets of rules which are useful and interesting for the user.

2 Virtual Reality Technology

Contents

2.1 Introduction

As previously already stated, visualisation and interaction are an important fields of research in Data Mining (DM). It relies on the fact that the human mind processes visual information easily and quickly and may extract a lot of information and knowledge in this way. In this context, visualisation represents a critical step that is required at during the Knowledge Discovery in Databases (KDD) process (Fayyad *et*

al. 1996 [101]).

As proposed by Fayyad *et al.* 1996 [281], data visualisation can be viewed as an approach to the problem of KDD and involves two important aspects (Fayyad *et al.* 1996 [101]): (*i*) helping the user in the knowledge discovery step and (*ii*) enabling him/her to interact with the knowledge representation (Chernoff 1973 [68], Bertin 1967 [24], Card *et al.*1999 [60], Becker *et al.* 1987 [235], Pickett and Grinstein 1988 [220], Tufte 1990 [278], Cleveland 1993 [74], Keim and Kriegel 1996 [167], Wong and Bergeron 1997 [296], Friendly 2001 [110], Unwin 2000 [279]). Thus, VDM requires tools and efficient visualisation and interaction techniques enabling the user(s) to perform such complex tasks.

The emergence of virtual reality (VR) has enabled significant advances in the field of data visualisation (large-scale, stereoscopic viewing) and user interaction (Fuchs *et al.* 2003[112]). VR allows immersive real-time visualisation and interaction with huge amounts of data. Therefore, VR has a great potential and can be very useful for VDM. However, current VR systems (many configurations exist), interaction techniques and metaphors are not well suited for the specific tasks involved or required by VDM and KDD.

In the first part this chapter, we define the concept of VR and Virtual Environments (VEs). Then, we compare current 2D, 3D and VR techniques to now proposed in the context of data visualisation. In Section 3, we present and analyse the various interaction devices and interfaces widely used in current VR applications. In addition, we review existing 3D interaction techniques and metaphors and give the advantages and drawbacks of these for DM, both in mono-user and multi-user contexts. Finally, we propose a classification for hardware configurations of visual displays enabling user immersion in VEs.

2.2 Concepts and definition of VR

Virtual reality lies at the intersection of several disciplines such as computer graphics, human-computer interaction, computer vision, and robotics. It uses large-scale advanced visual displays, interaction devices and multimodal interaction techniques to immerse one or more user(s) in a Virtual Environment (VE). One the advantages of VR is that the interaction techniques are based on human natural action, perception and communication abilities, allowing the user to perform complex tasks quite intuitively (Burdea and Coiffet 1993 [54], Fuchs *et al.* 2003 [112])

VR is currently used in a wide variety of applications such as engineering, medicine, biology, education and training, etc. The increasingly widespread use of VR is due to (*i*) the rapid evolution of computers (processing and graphic capabilities), interaction devices coming recently from the game industry such as the Wiimote^{TM}, Kinect^{TM}, Play Station Move^{TM}, and low-cost visual displays (3D televisions and projectors),

and (*ii*) the increasing development of applications in different sectors (automotive, aerospace, medicine, education, defence, entertainment, etc.).

Augmented Reality (AR) refers to different techniques allowing the integration of digital entities (objects, images, text, etc.) in the real world [196]. Milgram and Kishino [197] coined the term *Mixed Reality* and proposed to consider it as a continuum which extends from the real world to fully virtual worlds (Figure 2.1).

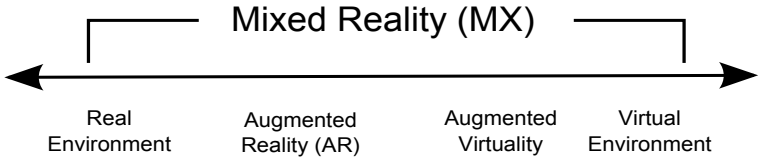


FIGURE 2.1: Triangle of Virtual Reality proposed by Burdea and Coiffet [54].

Burdea and Coiffet 1993 [54] introduced the *3I* concept: *immersion*, *interaction* and *imagination* as a new approach to define and analyse VEs (Figure 2.2). In this concept, *Imagination* plays a crucial role and characterises user interpretation that results from a VR experience. Some authors consider VR as an extension of classical human-computer interfaces. For example, Ellis 1994 [97] proposed that VR is "an advanced human-computer interface that simulates a realistic environment and allows participants to interact with it". Other researchers define VR as computer simulated worlds in which the user is the main actor.

In this context, it is important to understand that a user may perceive and understand a VE through various sensory channels such as vision, sound, touch, smell, etc., but also through movement, action and interaction with virtual objects or data. Tisseau 2001 [274] defines VR as a universe of models offering the mediation of senses, actions and mind.

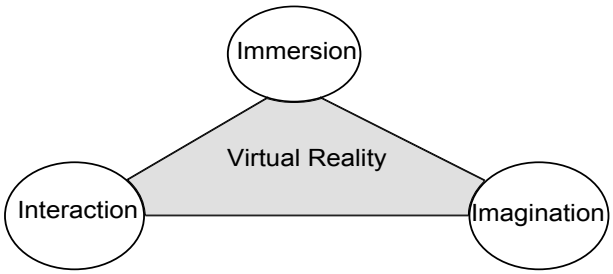


FIGURE 2.2: Triangle of Virtual Reality proposed by Burdea and Coiffet[54].

Other definitions of VR from functional, technical, and philosophical points of view are detailed in Fuchs et*al.* 2006 [113] and Fuchs et*al.* 2003 [112]. Fuchs *et al.* [112] define VR through its purpose: "the purpose of VR is to enable a person (or more) a sensory-motor and cognitive activity in an artificial world, numerically

created, which can be imaginary, or a symbolic simulation of certain aspects of the real world".

Zeltzer 1992 [307] has proposed a model for the description and classification of VEs. The model, represented by the AIP cube, includes three components: *Autonomy*, *Interaction* and *Presence* (Figure 2.3) and is based on the assumption that every VE has three distinct components which are:

- A set of models and processes (*autonomy*);
- Action on them (*interaction*);
- Some sensory modalities (*presence*).

Using this model, any VR application can be represented using three-dimensional coordinates (*autonomy*, *interaction*, *presence*), for which the axes are normalised between 0 (criterion completely absent) and 1 (criterion fully present). Tisseau 2001 [274] also focused on the concepts of *presence* and *autonomy* to define VR applications and propose that the *presence* itself is supported by *immersion* and *interaction* (Figure 2.4).

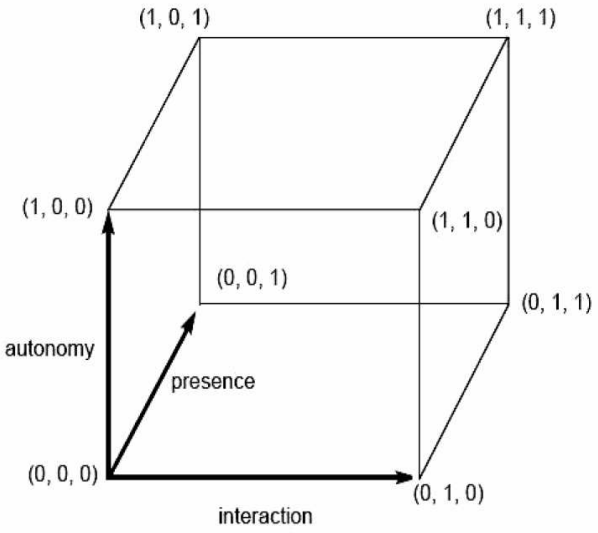


FIGURE 2.3: The AIP cube : *autonomy*, *interaction*, *presence* [307].

2.2.1 Immersion

In the VR community, the term *immersion* is widely used and it is quite important to understand what it really means and what are the related aspects. The first level of

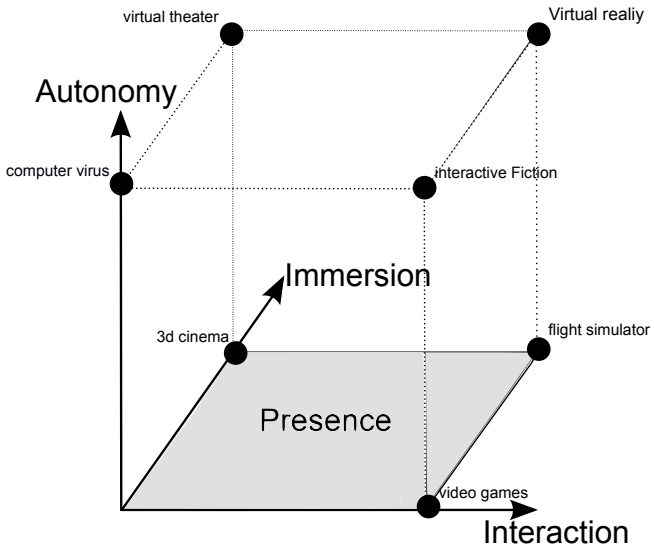


FIGURE 2.4: *Immersion*, *interaction*, and*autonomy* in VR [274].

immersion is physical or visual (Burkhardt 2003 [56], Mestre and Fuch 2006 [194]). In fact, *immersion* reflects the extent to which the human visual field is covered by the displayed virtual images. The second level of *immersion*, generally called spatial immersion, is related to the extent to which the user(s) can act or move in the virtual world. These two levels of *immersion* are of course closely linked.

Bowman and Hodges 1999 [39] define *immersion* as the sensation of being present in the VE. Thus, a user is physically "immersed" when he/she feels that the physical world that surrounds him/her has replaced the virtual world. A further level of *immersion* is related to the user's mental or psychological state. For some VR applications such as training or simulation, this level of immersion is required.

The user's sense of presence in a virtual world is another important aspect that plays a crucial role in immersion. Indeed, the presence provides the user the feeling of being inside the VE. In this context, it is essential to distinguish between virtual presence and social presence (Slater *et al.* 1998 [252], Slater et*al.* 1996 [253]). The first type of presence is the feeling that we try to give to the user as he/she is a part of the virtual world (Burkhardt 2003 [56], Mestre and Fuchs 2006 [194], Stoffregen *et al.* 2003[265]). On the other hand, social presence characterises collaborative VEs in which virtual humans (human avatars or embodied conversational agents) are present. Social presence involve mutual awareness of the participants and of their activities within the shared virtual world. Kadri *et al.* 2007 [162, 163] point out the importance of user's representation by avatars in the VE.

2.2.2 Autonomy

In VR, the notion of *autonomy* is related to the different components of the VE. The user is one of those components and he/she is considered as the most active entity in the virtual space. The user role in VR is not the same as in scientific or interactive simulations (Tisseau 2001[274]). In scientific simulation, the user sets some parameters before the simulation and analyses the results afterwards. On the other hand, in VR applications, the user is exposed to a digital environment which allows him/her to evolve, to change the VE properties, and to interact with the different entities in real time. The user's *autonomy* lies in his ability to coordinate perceptions and actions taken during the interaction process.

2.2.3 Interaction

Since the emergence of VR, researchers have been particularly interested in *interaction*, this being is the core component of any interactive system. Broadly speaking, *Interaction* can be defined as a communication language between humans and computers. This communication language is the set of actions/reactions between human and computer interfaces through input / output devices and interaction techniques.

3D interaction can be defined as a system that links different interaction devices and software technology to enable real-time modification of VEs. These different pieces of software allow the use of available materials through drivers which provide access to low-level devices, and high-level software applications. The different interaction techniques are between the hardware layer (low-level) and the application layer (high-level).

2.3 Virtual Environments

The term *Virtual Environment* was introduced by researchers from the Massachusetts Institute of Technology (MIT) in the early 90s as a synonym for VR (Heim 1995 [140]). A VE is generally considered as a 3D real-time interactive environment representing real, symbolic or imaginary data, visualised using desk-top or immersive displays (Hachet 2003 [132]). There are different types of VE, based on the degree of immersion provided to the user (Kalawsky 1996[164]):

- Non-Immersive Virtual Environment (NIVE)
- Semi-Immersive Virtual Environment (SIVE)
- Fully-Immersive Virtual Environment (FIVE)

Table 2.3 shows the qualitative performance of the different types of VR systems according to the degree of immersion provided by the associated VE.

In VR, users need to interact with the virtual objects present within the VE using some interaction techniques. An interaction technique is defined as a method enabling a task in a VE to be performed (Hachet 2003 [132]). Foley *et al.* [103] consider interaction technique as a way of using 3D input devices to accomplish a given task. Interaction techniques can also be defined as how the user operates a given interface to interact in real-time with virtual entities and control the application. In VR, the notion of interaction paradigm or metaphor is used by some authors to define a set of rules and techniques that allow the user to perform interactive tasks within a VE.

Main features	NIVE	SIVE	FIVE
Resolution	High	High	low-average
Perception	Low	Low	High
Navigation skills	Low	Average	High
Field of view	Small	Average	Large
Shift	Low	Low	High-average
Sense of immersion	Low	High-average	High-average

TABLE 2.1: Qualitative performance of the various VEs [164].

2.4 From 2D toward 3D and Virtual Reality

There is a controversial debate on the use of 2D versus 3D and VR for information visualisation. In order to justify our choice for 3D and VR, we first review the difference between 3D visualisation and VR techniques:

- 3D visualisation is a representation of an object in a 3D space by showing length, width and height coordinates on a 2D surface such as a computer monitor. 3D visual perception is achieved using visual depth cues such as lighting, shadows and perspective;
- VR techniques enable the user immersion in a multi-sensorial VE and use interaction devices and stereoscopic images to increase depth perception and the relative 3D position of objects. In addition the user is able to navigate and explore VEs.

2.4.1 2D versus 3D

Little research has been dedicated to the comparison of 2D and 3D representations. The experiments of Spence 1990 [255] and Carswel *et al.* 1991 [62] show that there is no significant difference of accuracy between 2D and 3D for the comparison of

numerical values. In particular, Spence 1990 [255] pointed out that it is not the apparent dimensionality of visual structures that counts but rather the actual number of parameters that show variability. Under some circumstances, information may be processed even faster when represented in 3D rather than in 2D. Concerning the perception of global trends in data, the experimental results of Carswel *et al.* 1991 [62] also show an improvement when answer times using 3D but to the detriment of accuracy.

Finally, Tavanti and Lind 2001 [270] pointed out that realistic 3D displays could support cognitive spatial abilities and memory tasks, namely remembering the place of an object, better than with 2D.

Cockburn and McKenzie 2002 [75] presented their investigation in the domain of effectiveness of spatial memory in real-world physical models and in equivalent computer-based virtual systems. The different models vary the user's freedom to use depth and perspective in spatial arrangements of images representing web pages. Six interfaces were used in the evaluation of three physical interfaces and three computer-based virtual equivalents. Results show that the ability to quickly locate web page images deteriorated as their freedom to use the third dimension increased. The subjective answers also indicated that the users found the 3D interfaces more cluttered and less efficient. Spacial memory clearly provides an effective aid to information retrieval, but the study doesn't conclude about the role that 3D plays to increase rapidity of data retrieval in static-perspective spatial organisations. The results indicate that for relatively sparse information retrieval tasks (up to 99 data items), 3D hinders retrieval.

Tory *et al.* 2006 [276] compared 2D, 3D, and combined 2D/3D displays for different tasks in order to identify the tasks for which each view is best suited trough 3 experiments: (*i*) position estimation task, (*ii*) orientation task, and (*iii*) qualitative exploration. These experiments showed that strict 3D displays with additional cues such as shadows can be effective for approximate relative position estimation and orientation. However, precise orientation and positioning are difficult with a strict 3D display. For each precise task, the combination of 2D/3D display was better than strict 2D or 3D display. Compared to strict 2D display, combination displays performed as well or better.

On the other hand, several problems arise, such as intensive computation, more complex implementations than 2D interfaces, and user adaptation and disorientation. The first problem can be addressed by using powerful and specialised hardware. However, one of the main problems of 3D applications is user adaptation. In fact, most users only have experience with classical windows, icons, menu pointing devices (WIMP) and 2D-desktop metaphors. Therefore, interaction with 3D presentations and possibly the use of special devices require considerable adaptation efforts to use this technology. There is still no commonly-accepted standard for interaction with 3D environments. Some research has shown that it takes users some time to understand what kind of interaction possibilities they actually have (Baumgartner *et al.*

2007 [17]). In particular, as a consequence of a richer set of interactions and a higher degree of freedom, users may be disoriented.

2.4.2 Toward Virtual Reality

Research on human visual perception attests to the complexity and power of human visual abilities. We perceive a 3D world primarily through a combination of binocular vision and the use of motion parallax (Harris 2004 [138]). However the brain obtains many clues such as the depth and shape of objects from the surrounding environment (ground-plane, shadow, relative motion). Immersive environments allow to a user to take advantage of the way the brain already interprets visual information (Hubona and Shirah 2005[149]) and provides key advantages for evaluating and analysing information, including the use of peripheral vision to provide global context, body-centric judgements about 3D spatial relations, and enhanced 3D perception from stereo and motion parallax (head tracking, Dam et*al.* 2000[86]).

Together these spatial indicators create a more natural environment and thus promote more efficient exploration of 3D data. On a regular computer screen, the strongest available depth cue is motion parallax. As a result, in order to understand 3D data from images on a screen, users need to constantly rotate the view or rock it back and forth to perceive depth. The motion interfaces with detailed examination or measurement of displayed data is not required in stereoscopic environments where other depth cues are available. Furthermore, immersive environments offering head tracking in addition to stereoscopy enable motion parallax without the user having to consciously move the 3D data: motion is simply created by moving one's head or walking around the data; an intuitive response that does not interfere with analytical tasks.

To overcome limitations of interaction with 3D representations, VR interfaces and input devices have been proposed. These interfaces and devices offer simpler and more intuitive interaction techniques (selection, manipulation, navigation, etc.), and more compelling functionality (Shneiderman 2003[245]). Latulipe *et al.* 2005 [176] demonstrated that a symmetric technique for simultaneous rotation, translation and scale yields significant performance benefits over the standard single mouse in an image alignment task.

In VR, the user can always access external information without leaving the environment and the context of the representation. Also, the user's immersion in the data allows him to take advantage of stereoscopic vision that enables him to disambiguate complex abstract representations (Maletic *et al.* 2001 [190]). Ware and Franck 1996 [288], compared the visualisation of 2D and 3D graphs. Their work shows a significant improvement in *intelligibility* when using 3D. More precisely, they found that the ability to decide if two nodes are connected or not is improved by a factor of 1.6 when adding stereo cues, by 2.2 when using motion parallax depth cues, and by a factor of 3 when using stereoscopic as well as motion parallax depth cues.

Aitsiselmi and Holliman 2009 [6], found that the participants obtained better scores if they were doing a mental rotation task on a stereoscopic screen instead of a 2D screen. They found that the participants obtained better scores if they were doing a mental rotation task on a stereoscopic screen instead of a 2D screen. This result demonstrates the *efficiency* of VR and shows that the extra depth information given by stereoscopic display makes it easier to move a shape mentally.

We can therefore conclude that stereoscopy and interaction are the two most important components of VE and the most useful to users. Therefore, the equipment used should be taken into account from the very beginning of application design, and consequently be taken into account as a part of VDM technique taxonomy.

2.5 Interaction techniques and metaphors

Since the very beginning, researchers have been interested in 3D interaction which can be regarded as the main component of any VR system. The information visualisation community has begun to distinguish between low-level interaction (between the user and the software interface) and high-level interaction (between the user and the information space). In low-level interaction, the user's goal is often to change the representation to uncover patterns, relationships, trends or other features. Amar *et al.* 2005 [7] define a set of low-level tasks that are typically performed with visualisation. These primitive tasks are to retrieve values, filter, compute derived values, find extremes, sort, determine ranges, characterised distribution, find anomalies, cluster and correlate; these all accommodate specific questions that might be asked of a visualisation, and they can be composed into aggregate functions for more complex questions.

The P-set model, proposed by Jankun-Kelly *et al.* 2007 [159], offers an approach for capturing a user's sequence of low-level interactive steps in an exploratory visualisation system. Tracking the investigation process allows the user to see their current state in the context of prior exploration and can potentially inform future actions. Tools such as the ones proposed by Palantir [1] are now implementing history mechanisms that expose the sequence of interactive steps as a sense making aid, and Aruvi and van-Wijk 2008 [246] integrate history tracking with diagrammatic knowledge capture. In high-level interaction, the user's goal is to generate understanding.

In this context, understanding the intent of the interaction becomes critical. Yi *et al.* 2007 [302] proposed a taxonomy of interaction intents: *select*, *explore*, *reconfigure*, *encode*, *abstract*, *filter* and *connect*, that could constitute the components of a knowledge discovery or confirmation process. Just as low-level interaction capabilities can be used to assess completeness of an interface (does it allow users to efficiently and effectively perform each low-level operation?), these higher-level categories can be used to assess the kind of goals to which an interface could be applied.

Traditional 2D user interfaces (*WIMP*) have well-established methods and techniques, such as pull-down menus, buttons, and windows. Generally, these methods cannot be transferred to VR. Therefore, VE needs completely new interaction paradigms. Three-dimensional user interfaces (3DUI) are considered to be one of the alternative of current 2D interfaces (Dachselt and Hinz 2005 [84]). More than a dozen years of research in the field of VR have produced a rich variety of applications, novel input and output devices as well as 3D interaction techniques.

There are several classifications of 3D interaction techniques. Mine 1995 [198] proposed the first classification based on four fundamental tasks: *navigation*, *selection*, *manipulation*, and *scaling*. He also defined a fifth task derived from the four previous tasks: *virtual menu* and *widgets*. Hand 1997 [137] introduced the modern classification basics which were taken over by Bowman *et al.* 2001 [41] who classifies the various 3D interaction techniques according to three main tasks: *navigation*, *selection* and *manipulation*, and *system control*. The authors also discussed the effect of common interaction devices on user interaction as well as interaction techniques for generic 3D tasks. Arns 2002 [11] considered that Bowman's taxonomy is too general and encompasses too many parts of a VR system. For that reason, she proposed a classification for virtual locomotion (travel and way-finding) methods. This classification includes information concerning visual displays, interactions devices, tasks, and the two primary elements of virtual travel: *translation* and *rotation*.

Coquillart and Grosjean 2003 [241] proposed an alternative classification of interaction techniques. They broke down each application into basic tasks called *Primitive Virtual Behavioural* (PVB). These PVBs are the functional objectives of immersion and interaction level. These basic tasks may of course be performed by the user through the proposed interaction techniques. Whatever the application, the PVB can be grouped into four categories: to *observe* the virtual world, to *move* in the virtual world, to *act* on the virtual world, and to *communicate* with others or with the application. Finally, Dachselt and Hinz 2005 [84] proposed a classification of 3D widget solutions by interaction purpose/intention of use, e.g, direct 3D object interaction, 3D scene manipulation, exploration and visualisation.

In the following, we present an overview of the three 3D interaction tasks involved in the Bowman's *et al.* 2001 classification [41]: *navigation*, *selection* and *manipulation*, and *system control*. These interaction primitives have been proposed in a general context of interaction with VEs and are not well suited for Visual Data Mining tasks. Thus, some specific primitives will have to be designed and integrated in interaction technique classification in this more complex context.

2.5.1 Navigation

As in the real world, the user needs to move in the virtual world to perform certain tasks such as exploring large 3D structures or datum sets. During navigation, the user may need to move his/her head to observe some specific objects or to better

perceive the 3D layout of the VE through motion parallax. *Navigation* is generally defined by a set of methods used to move the virtual camera (the user's view point) (Rheingold 1991 [233]). Some researchers (Dumas et*al.* 1999 [93]) defined *navigation* as all user's movements within the virtual space. For Bowman *et al.* 2004 [42], this task is the most relevant user action in large-scale 3D environments. It allows the user to browse, search and/or operate in virtual space. For example, Wiss and Carr 1999 [294] have performed a comparative study of three different 3D representations of information: cam tree, information cube, and information landscape. The results indicate that one of the most influential factors was *navigation*. *Navigation* may sometime be reduced to *locomotion* as the user has to propulse himself/herself in the VE using some physical activities such a walking, bicycling, etc. (Darken et*al.* 1998[87], Chance et*al.* 1998[66]).

Navigation presents challenges such as supporting spatial awareness, providing efficient and comfortable movement between distant locations so that users can focus on more important tasks. Bowman *et al.* 1997 [40] define two main components for *navigation*: the motor component called *travel*, and the cognitive component called *wayfinding*. *Travel* is the motor component of *navigation* and refers to physical movements of the user from one place to another. *Wayfinding* matches the cognitive component of navigation. It allows users to navigate in the environment and choose a movement path (Fuchs et*al.* 2003[112]). In this case, the user asks questions such as: "Where am I?", "Where should I go?" , "How do i get there?".

Several factors influence navigation techniques quality. Bowman *et al.* 1997 [40] define a list of quality factors which represent specific attributes of effectiveness for virtual travel techniques.

- Travel speed: appropriate velocity;
- Accuracy: proximity to the desired target;
- Spatial Awareness: the user's knowledge of his/her position and orientation in the virtual environment during and after navigation;
- Ease of learning: the ability of a novice user to take ownership of the navigation technique;
- Ease of use: the cognitive load of the technique from the user's point of view;
- Information gathering: the user's ability to collect information on the environment during the navigation;
- Presence: the users sense of immersion within the virtual environment.

Later, the same authors Bowman 1998 [44] proposed a taxonomy of travel techniques in VE. This taxonomy has 3 main components (Figure 2.5): direction/target

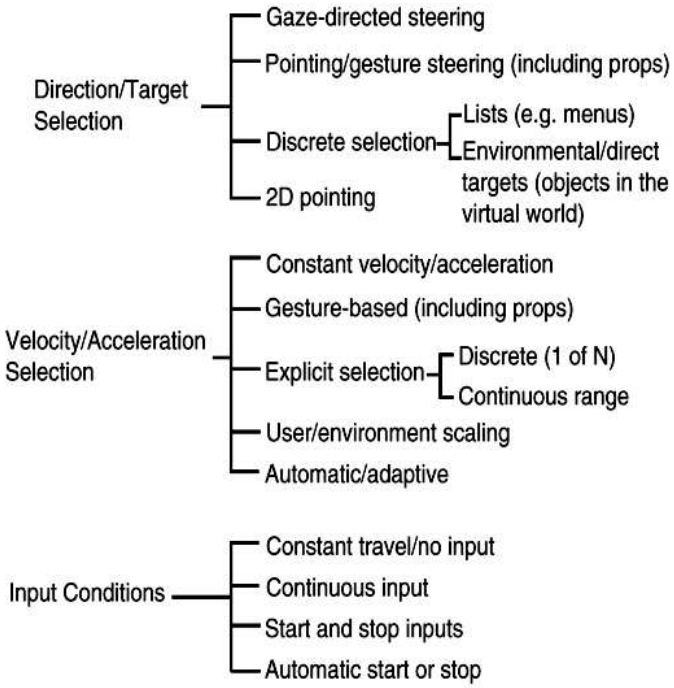


FIGURE 2.5: Bowman's taxonomy for travel techniques [44].

selection, velocity/acceleration selection and input condition.

- **direction/target selection**: in order to move in the VE, the user must indicate *where* he/she wants to go, or at least in what direction. In the first case, the user can simply select an exact target to move to, using a direct manipulation technique such as ray-casting, or an indirect technique such as selecting it in a menu. In the second case , the user may not specify an exact location, but instead a direction of travel, by pointing with a joystick or physically rotate in the desired direction for example;

- **velocity/acceleration selection**: once the user indicates where he/she is going, the designer should worry about how quickly the user will get there. Several techniques exist for determining the travel speed. The user can select a discrete speed (low, medium or fast) or a continuous rang (a slider bar or a "gas pedal"). Additionally, it is important to specify how the user reaches this speed. A sudden transition between no motion to fast travel speed will certainly disorient the user, so it may be beneficial to allow the user to accelerate or decelerate in a VE;

- **input conditions**: several input devices can be used to ensure the user travel in a VE. The user can simply indicate the target or continuously indicate the target direction. The first technique may be easier for a novice user but can cause the user disorientation and does not provide as much control as the first

option, if for example the user changes his mind about where he wants to travel. Input devices will be studied carefully later.

Other factors was studied in Bowman's taxonomy include task characteristics, environment characteristics, user characteristics, and system characteristics.
Arns and Cruz-Neira 2002 [11] think that Bowman's taxonomy is too general, and can include too many parts of a VR system. To remedy to this problem, they proposed a new classification system based on the display devices and interaction devices.

Under Bowman's taxonomy using a wand with constant velocity or an omnidirectional sliding device appear identical because the user moves at constant velocity in the direction of gaze. However, these two travel techniques lead to very different experiences for the user. In the same way, the choice of the display device may also affect the user's travel in a VE. The authors also define two sub-tasks that the user performs when travelling: translation and rotation. Both of these operations can be performed simultaneously or separately.

- **Rotation**: two rotation methods (Figure 2.6) are defined for the user of a VE : *physical rotation* and *virtual rotation*. Physical rotation is the rotation of the user with respect to the world. Virtual rotation is the rotation of virtual world with respect to the user. Usually, when travelling in a physical world, we travel forward in the direction our body is facing. When we wish to change direction, we need to orient our body in the new direction and move in that direction. However, in the real world, we are limited since we can't rotate around the vertical axis. In the VE, the user is still subject to the same limitation. However, he/she is able to fully rotate his/her body in the VE and rotate the VE with respect to his/her body.
- **Translation**: similar to rotation, view point translation can be accomplished using two different methods: *physical translation* and *virtual translation* (Figure 2.7). Physical translation can be simply walking to change the viewpoint. Because virtual worlds are often much more larger than the physical area in which the user walks (limitation due to the locomotion device), other methods may be employed such as a bicycle which allows the user to feel that he/she is physically walking as he/she translates in the virtual world, while actually remaining stationary in the physical world. Virtual translation is similar to virtual rotation. In this context, a variety of devices can be used such as a joystick.

Navigation is a conceptually simple task that involves a movement of the viewpoint from one location to another. Additionally, viewpoint orientation also needs to be considered. Bowman *et al.* 2001 [41] organises the navigation metaphors into five categories:

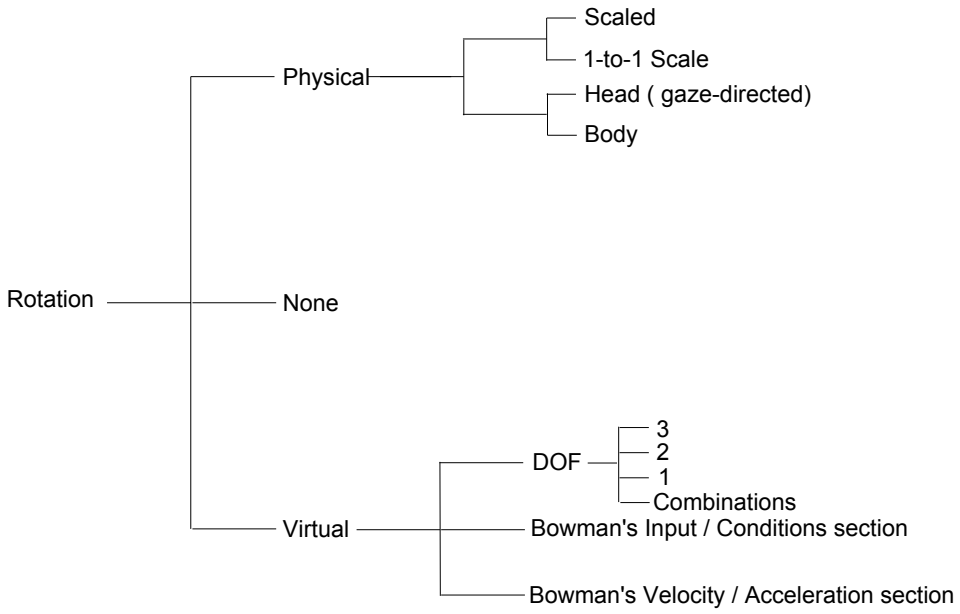


FIGURE 2.6: Arns's 2000 [11] taxonomy for rotation techniques.

- *Physical movement;*
- *Manual viewpoint manipulation;*
- *Steering;*
- *Target-based travel;*
- *Route planning.*

Physical movement

In most cases of 3D interaction, the concept used for the design of new interaction techniques is inspired by human interaction with the real environment. For example, walking is the easiest and most natural way to move from one place to another in everyday life. This way of moving is widely used in VR. Other navigation metaphors are also inspired by such natural dynamic gestures.

For example, physical movement uses the motion of the user's body to travel through the environment. Ware and Osborne [289] defined one of the first navigation metaphors based on real walking. The user moves freely within the VE by physically walking on the spot. To change the walking direction, the user should turn his head in the desired orientation. Figure 2.8 shows some examples of locomotion devices such as treadmills, stationary bicycles, walking-pads, dance-pads, and a chair-based interface. Such techniques need enhanced physical exertion when navigating.

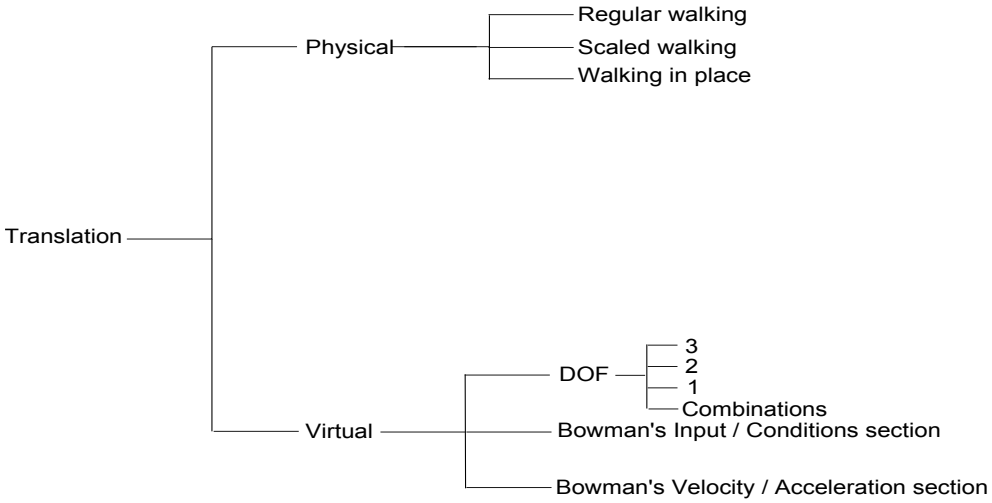
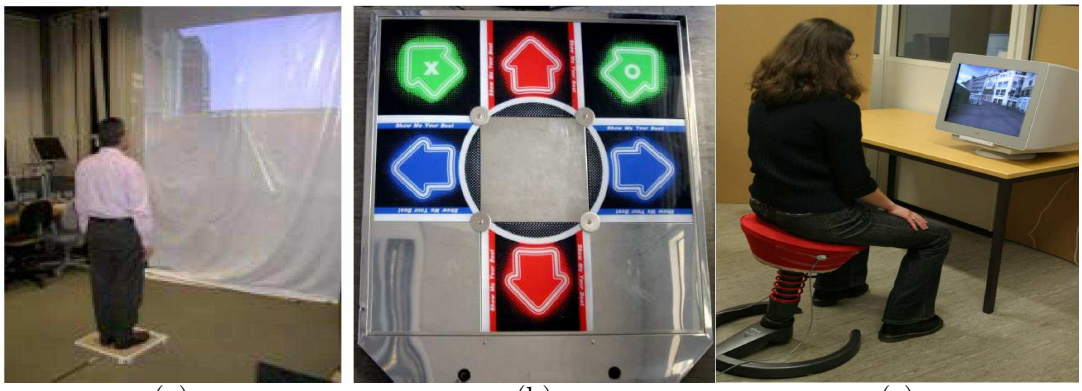


FIGURE 2.7: Arns's 2002 [11] taxonomy of translation techniques.



(a) (b) (c)

FIGURE 2.8: Examples of locomotion devices : (a): a walking-pad [35], (b): a dance Pad [22], and (c): a chair-based interface[22].

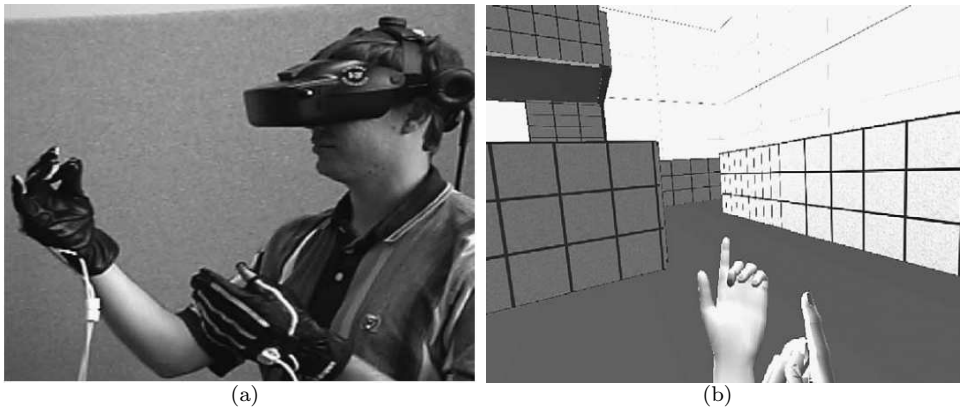
Another approach based on the user physical movement principle was proposed by York *et al.* 2004 [303]. They proposed to calculate the travel speed based on the speed of ascent or descent of the user knees. There is a strong dependence between the travel in the virtual scene and physical movement navigation techniques. The techniques based on the physical movement navigation are easier to use because the user does not provide any cognitive effort to understand this technique. According to a study by Usoh *et al.* 1999 [282] study, the results are better when the used navigation metaphor is close to real walking, because presence is higher for real walkers than virtual walkers.

Virtual movement

Manual viewpoint manipulation In manual viewpoint manipulation, the user's hand motions can be used to simulate travel. Mine *et al.* 1997 [199] introduced a travel

technique in which the direction is given by the orientation of the head. It means that the direction of movement follows the direction of the user's gaze which is determined by the movement of the head. It is a cognitively very simple technique but it has a major drawback since it does not allow the user to visually observe the parts of the scene that are behind him or to the side, so he will often be forced to navigate blindly. To overcome this problem, a technique based on the user hand direction was proposed (Robinett and Holloway 1992 [238], Bowman and Hodges 1999 [39]). This technique allows the user to move and look in different directions. The travel direction is determined continuously by the user's hand orientation. This technique is somewhat more difficult to learn for some users, but more flexible then the head tracking techniques. Mine *et al.* 1997 [199] proposed a technique which uses both hands to move. Indeed, it is possible to determine both the direction and the velocity of travel. The main advantage of this technique is based on the knowledge of the position of both hands. The velocity is calculated according to the distance between both hands: the longer the distance is, more rapid the travel is. This technique is cognitively difficult because the user may have difficulty to control the velocity of his travel.

The "grabbing-the-air" technique (Butterworth *et al.* 1992 [57]) is another example of techniques that use the hand to specify the travel direction. The user pulls himself along as if with a virtual rope. The direction of movement may be indicated in several ways, among them, using a joystick. In the metaphor of the flying saucer introduced by Butterworth *et al.* 1992 [57], the user uses a joystick to move forward and backward in the VE. There are 6 degrees of freedom device, 3 degrees for motion and 3 degrees for rotation. This technique is often performed using *Pinch Gloves* and can be used with one or both hands . Usually, when someone talks to others, it is natural to turn his body toward his interlocutor(s). Bowman *et al.* 2001 [36] were inspired by this real-life example to implement the technique that determines the travel direction with the user torso direction (Figure 2.9).



(a) (b)

FIGURE 2.9: *Pinch Gloves* [36]: (a): User wearing Pinch Gloves (b): Two-handed navigation technique

Target-based travel The user specifies the destination, and the system handles the

movement. This may take the form of *teleportation*, where the user jumps immediately to the new location. This method has a major drawback because it confuses the user since it gives no information on the traveled distance. To overcome this problem Butterworth *et al.* 1992 [57] proposed to fly the user between the starting point and the destination to avoid user disorientation. Target-based techniques are very simple from the user's point of view.

The use of a map is another way to perform a movement in a virtual world proposed by Bowman *et al.* 1999 [39]. The user is represented by an icon in a 2D map (Figure 2.10). The movement of the icon by a stylus to a new location on the map creates the user's travel. When the icon is pressed, the system moves slowly from the user's current location to the new location indicated by the icon.

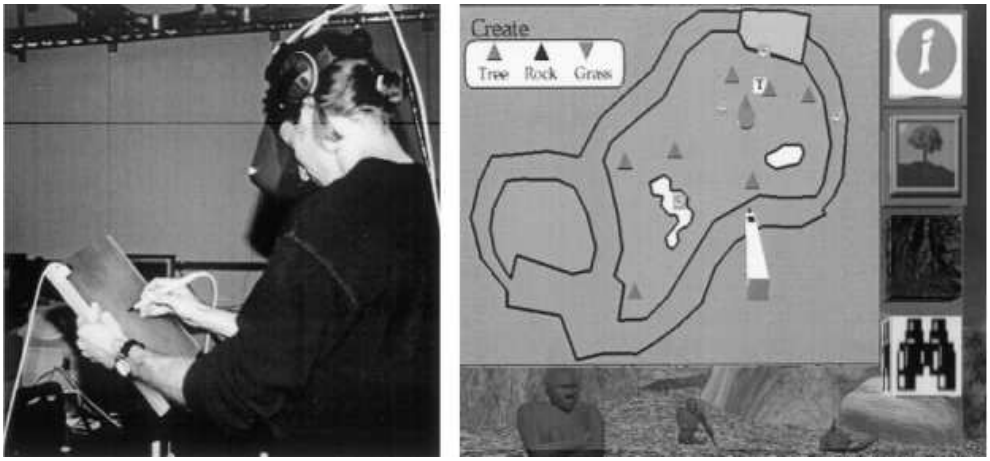


FIGURE 2.10: Physical (left) and virtual (right) view of map navigation metaphor [39].

Stoakley *et al.* 1995 [264] offered the user the possibility to directly manipulate his viewpoint. The user can move his viewpoint as he/she moves an object in the virtual world. This metaphor is called *Worlds In Miniature* (WIM). To travel, the user holds in his hand a virtual model (a miniature representation of the virtual world). The user's movements in the virtual world following his/her avatar in the miniature world. Elvins *et al.* 1998 [98] propose to represent only a part of the virtual world in miniature allowing the user to easily find its way. This technique relies on the principle of *Worldlets* that are miniature representations of several parts of the virtual environment.

Route planning The user specifies the path that should be taken through the environment, and the system handles the movement (Ahmed and Eades 2005[5]). The user may manipulate icons or draw a path on a map of the space or in the actual environment in order to plan a route. Igarashi *et al.* 1998 [150] proposed another method that allows the user to draw the intended path directly on the scene, and the avatar automatically moves along the path. Sternberger 2005 [263] uses a radius

to draw freely a deformable virtual path to follow during navigation. The purpose of this interaction is to avoid obstacles that confront the user and that can slightly change the direction. Once the destination is known, the user can be moved to the new location.

Steering Continuous specification of the direction of motion. Examples of techniques called gaze-directed steering, where the user's head orientation determines the direction of travel, or pointing, where hand orientation is used are called continuous steering; these techniques are general and efficient.

Thought wizard metaphor The user is relaxed at a fixed point in the 3D space and uses simple gestures to move the needed information around him or her.

2.5.2 Selection and manipulation

Because direct manipulation is a main interaction modality, not only in the 3D virtual world but also in natural real environments, the design of interaction techniques for object selection and manipulation has a profound impact on the entire VE user interface quality.

select an object is a common task in everyday life. Indeed, in order to manipulate an object, the user needs to take it in his hand or designate it among other objects. The selection process in VR is often inspired by the selection in the real world. The selection task is also called a *target acquisition* task (Zhai et*al.* 1994[308]), represents the designation of an object or set of objects to accomplish a given goal within the VE (Bowman and Frohlich 2005[38]). But how can one tell the system that an object has been selected?

Selection validation is the task after the designation task. It can be indicated in several ways according to the selection technique used and the environment in which the user operates. For example, the user can press a button, use a gesture or a voice command, but the validation can be done automatically if the interaction system takes into account the users intentions.

Navigation and selection are tasks that allow humans to have the illusion of living in a virtual world, to travel within it, and even to touch the objects belonging to the virtual world. In most cases, the user remains a spectator which is immersed in the VE. However, the manipulation task allows to the user to be an actor capable of changing the VE properties. It represents the active component of any interactive system. It can be defined as a complex process of modifying the properties of an object or set of objects belonging to the virtual world. These properties can be for example: position, orientation, colour, scale, and texture.

The manipulation task is related to the selection task, because the user can not

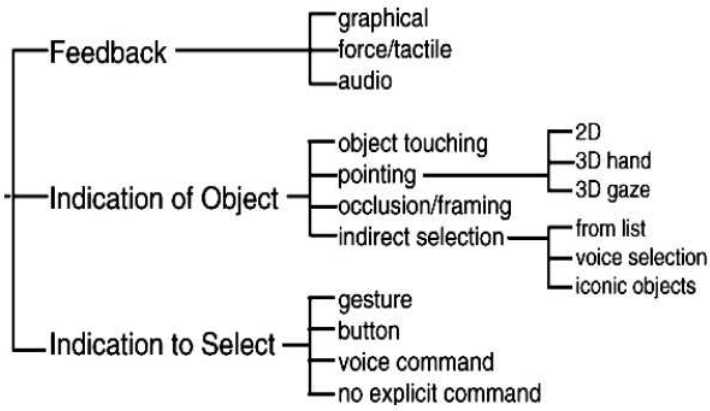
manipulate an object without having previously selected it. Figure 2.11 present taxonomies proposed by Bowman for selecting and manipulating virtual objects (Bowman 1998[44]). Interaction techniques for 3D manipulation in virtual environments should provide means of accomplishing at least one of the 4 basic tasks:

- Object selection;
- Object positioning;
- Object rotation;
- Object scaling.

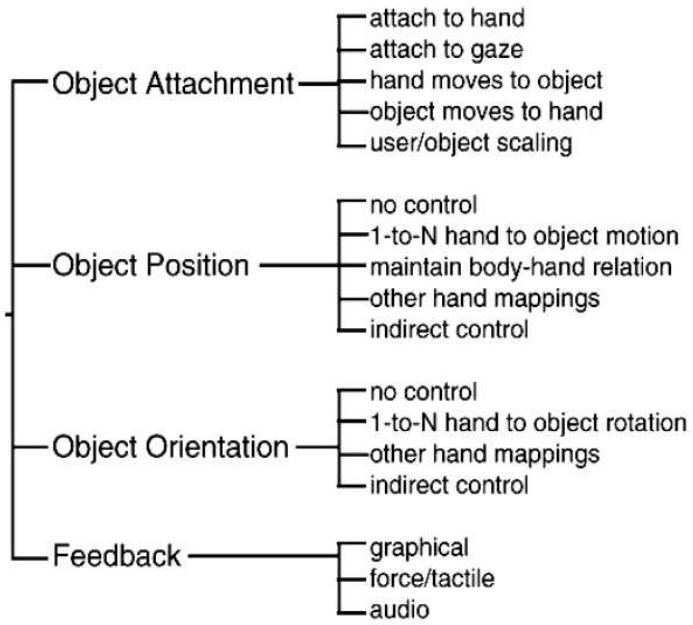
There are two main category techniques of selection and manipulation depending on the position and the distance of the user and virtual objects (Poupyrev and Ichikawa 1999 [228]): *Exocentric* techniques and *Egocentric* techniques (Poupyrev *et al.* 1998[226]).

***Exocentric* techniques**
In this technique, the virtual world is controlled from outside. In this case, the user is considered as an actor who is not a part of the virtual scene, but still has the power to act on objects in the virtual world. Stoakley *et al.* 1995 [264] proposed one of the first selection metaphors based on the principle of the exocentric interaction, called WIM, which uses miniature representations of the virtual scene to allow the user to act indirectly on the virtual world objects. Each object in the WIM can be selected using the virtual hand metaphor. The user holds a model of the scene on his/her non-dominant hand and selects (and/or manipulate) objects with his/her dominant hand. The main disadvantage of using the miniaturised model of the virtual world is the selection and manipulation of small objects. Also, all objects should be represented in the WIM even where the number of objects presented is very large. Pierce *et al.* 1999 [222] proposed the *Voodoo doll* technique. This technique offers the user the ability to create their own miniature objects which are called *Dolls*. To manipulate the objects, the user designates the object he/she wishes to handle due to the *head crusher* technique (Pierce *et al.* 1997 [221]). Then, a miniature model of the object and its immediate environment is created in the non-dominant hand. The dominant hand is used to move and rotate the miniature created. This technique allows the manipulation of objects of various sizes.

***Egocentric* techniques**
In this technique, the user proceeds directly from within the environment. The user can use his own hand to select a virtual object. Sturman *et al.* 1989 [266] proposed a selection technique based on the *virtual hand* metaphor. In this technique, the user touches the virtual object with his real hand to designate it, then validates the selection, either by closing his/her wrist, or by remaining in contact with the object for some time. This technique is very simple, natural and intuitive but it raises



(a)



(b)

FIGURE 2.11: Taxonomies proposed by Bowman 1998 [44] for selection (a) and object manipulation (b) in VEs.

the problem of selecting distant objects. In this case, the user must move up to be next to the object in order to designate it by his hand. To overcome this problem, Poupyrev *et al.* 1996 [227] proposed the Go-Go technique. It is based on the same principle as the previous technique, e.g. touching the virtual object to select it. It is also based on the metaphor of the virtual hand (the hand is represented by a real hand in the VE). The virtual hand position is calculated by a non-linear function, so

that the virtual hand goes further than the real hand after having reached a certain distance *D threshold*. The user has a virtual arm longer than its real hand allowing it to reach distant objects. This technique is still limited to the selection and manipulation of small objects. Frees and Kessler 2005 [106] have proposed a technique of selection/direct manipulation that can be used to complement manual interaction techniques such as *ray-casting* to improve the accuracy of user hand movements. In fact, the rotation and translation are stabilised when the hand movements slow down below a certain threshold.

When the virtual objects are not readily available to the user, he/she can select it through a specific interaction technique. One of the first techniques designed for interaction with virtual worlds is the *ray-casting* technique. This technique was introduced by Bolt 1980 [32] and enriched over the years by other researchers. The *Ray-casting* technique is based on the *virtual ray* metaphor. An infinite laser ray from a virtual hand crosses the entire virtual world. The first object intersected in the virtual world is ready to be selected. Zhai *et al.* 1994 [308] proposed a new interaction technique based on the *ray-casting* technique. They added a 3D semi-transparent cursor at the end of the ray. The objective of this cursor is to distinguish the virtual ray in the scene. Later, De Amicis *et al.* 2001 [88] replaced the cursor by a spherical volume.

Techniques based on the virtual pointer metaphor have the advantage of being cognitively simple and easy to use, but have a major drawback for the selection of small and distant objects. Liang and Green 1994 [183] proposed to use an icon instead of the ray to solve this problem. In fact, if distant objects become smaller with distance, then the selection tool must be larger to be able to easily select it. Thereafter, Forsberg *et al.* 1996 [104] proposed to modify the opening angle of the cone as a function of the object to be selected and its position in the virtual environment. The selection cone must be wider for distant objects that for close ones. This technique takes advantage of Fitts'law, which says that the selection time decreases with an increasing surface to be selected. During the selection process, the user may face barriers that hide the objects he/she wants to select. To avoid this difficulty, Olwal and Feiner 2003 [212] proposed the virtual flexible pointer technique which is an extension of the virtual ray technique. This technique allows a user in a 3D environment to point more easily to fully or partially obscured objects, and to indicate objects to other users more clearly. The flexible pointer can also reduce the need for disambiguation and can make it possible for the user to point to more objects than with other presented egocentric techniques (Figure 2.12).

Other researchers prefer to remove unwanted objects that the user does not wish to select (Steed and Parker 2004[261]). To do this, the user is holding a flashlight that illuminates some virtual objects, which are considered potentially selected. Unwanted objects can be removed by making movement with the lamp. This selection technique is effective and avoids selection errors but it has a major drawback, since the user changes the properties of the environment by removing undesirable objects to select other objects.

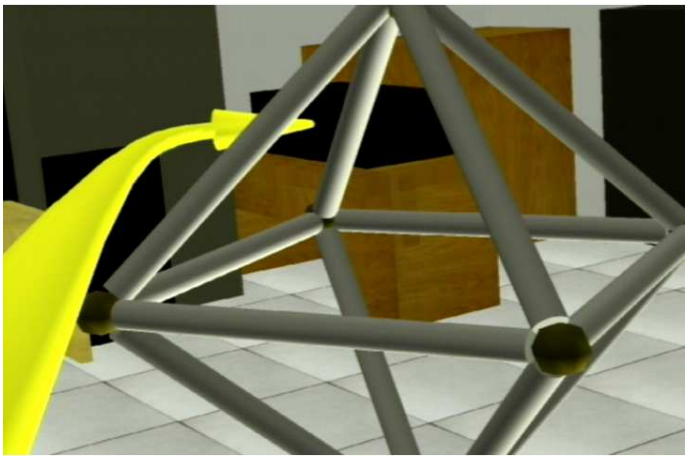


FIGURE 2.12: The flexible pointer selecting a partially occulted object without interfering with the occluding object [212].

For the different manipulation techniques that we have cited, the user acts directly on the virtual objects. There are, however, other techniques of indirect selection which allow the user to designate distant objects. These techniques are based on the *Directed Finger* metaphor (Pierce et*al.* 1997[221]). This metaphor requires that the index finger is recognisable in the virtual space using a position sensor attached to the finger called *Sticky Finger*. Objects are selected using a virtual ray, leading from the head of the user and passes through his hand index finger. Another technique based on the same metaphor was proposed by Tanriverdi and Jacob 2000 [269]. In this technique the user indicates objects with his glance. The user's head should be recognisable in the virtual world. This technique has a major problem because the user can not look around himself during the selection process. Pierce *et al.* 1997 [221] proposed an alternative selection technique that uses the movement of a hand. The user uses his thumb and forefinger to grasp the target object and take it, as if the image was perceived rather than the 3D object. Other techniques based on the same principle have been proposed. Among them, the *Framing Hands* technique (Pierce et*al.* 1997[221]). Using this technique, the user positions his hands to form the two corners of a frame in the 2D image. The user then positions this frame to surround the object to be selected. Another example of indirect manipulation techniques is the technique proposed by Kitamura *et al.* 1999 [169] using *Virtual Chopsticks*. This technique allows the user to capture, to move and to rotate virtual objects. For example, one of the chopsticks can serve as a rotation axis while the other indicates the rotation amplitude. Ware and Osborne 1990 [289] proposed the creation of new objects with shapes similar to virtual objects, on which they perform manipulations. The user has consistent tactile feedback. Hachet 2003 [132] uses a 2 hand-held steering wheel device (CAT), to manipulate objects in a scene.

The variety of reported interaction techniques can be overwhelming for the developer. However, some general principles regarding the choice of manipulation techniques can be stated. None of the techniques can be identified as the *best*. Their performance is task and environment dependent. Often, non-realistic techniques have better performance than those based on the real world (Bowman *et al.* 2001[41]). The evaluation of virtual manipulation techniques helps to quantify performance and is an important research area.

2.5.3 System control

System Control is a task that can execute a command in order to change the interaction mode and/or the system state. System control refers to indirect manipulation; it includes all indirect manipulation techniques on the application, the environment and/or data. Bowman *et al.* 2006 [37] define the *System Control* as the change of the state system or the mode of interaction; it can allow adjusting scalar values too. This task is at a different conceptual level from the three other tasks. The user interacts with the system using services provided by the system itself.

In a 2D interface, the *System Control* can be summarised with a simple click on an icon or menu. It can be considered as a communication tool between a human and the application. Well-known 2D techniques have been adapted to VE. In the 3D interface, the user must consider several degrees of freedom to interact with the system. Input/output devices are numerous and more elaborate than those used in 2D. Since the arrival of the first computers, graphic interfaces have evolved considerably. Currently, they are more ergonomic, more aesthetic, and easier to use. Early works reflection on control system techniques in VR proposed to extend or adapt some 2D widgets to 3D. Conner et al. 1992 [78] defined a widget as "an encapsulation of geometry and behaviour used to control or display information about application objects". Then, they built a library of components such as the colour, selector, and the virtual sphere rotation widget.

2.5.3.1 2D solutions in 3D environments

The achievement of a 2D menu in 3D interfaces started with the introduction of WIMP elements into VE such as pop-up menus and pull-down 3D virtual menus. Jacoby and Ellis 1992 [157] suggested a *pop up* 2D menu freely positioned and rotated in the virtual space. The user selects and activates the menu using the virtual pointer metaphor. This concept has been revised and improved by Jacoby et al. 1994 [158] by adding transparency and haptic feedback to facilitate the menu manipulation. By combining 3D interaction with the software support available for a 2D user interface tool, the user is provided with a familiar interaction concept. Other work attempted to make 2D widgets available within a 3D context, thus also incorporating traditional 2D menus. Early work by Feiner et al. 1993 [102] proposed a heads-up window system, where images of 2D windows are overlaid on the user's view of the

physical world. In Coninx et al. 1997 [77] an hybrid 2D/3D user interface is used. A pinch glove is used for simulating 2D mouse events. More recently, Andujar et al. 2006 [9] suggested to extend current 3D toolkits to display menus and other widgets such as 2D shapes within the virtual world.

In 2D desktop environments, menus can be operated by pressing a cursor key or using the mouse pointer. Typically in VE, the user's finger or laser-pointer is used for selecting in combination with a button-click on a physical device for activation. This solution has the advantage of being familiar to users. No learning period is necessary to teach user(s) how to use the menu.

2.5.3.2 3D menus

Different menus for system control have been proposed and evaluated:

Circular menus: are widely used in VRs. The different elements of the menu are placed on a circle. The selection is made either by motion in the direction of an element from the centre, or by circle rotation in order to bring the element of interest in the area of selection. This concept was firstly used in 2D interfaces. Kurtenbach and Buxton 1994 [175] proposed a technique that selects a menu item in a circular motion in one gesture. The user draws a line in the desired element's direction. This concept was then adapted for triangular menus whose elements are distributed around a central element. Thus manipulation is more efficient compared to the circular menus because the elements are equidistant from the centre. Deering 1995 [90] provided a significant improvement to these menus, adapting them to be hierarchical and then allowing a greater number of commands. The disadvantage of the hierarchical menu is that they occupy large a space on the screen. It completely replaces the scene during handling. Liang and Green 1994 [183] use circular menus without using hierarchy. Unlike other circular menus, the menu items are arranged in a circle around a vertical axis. The menu is then represented as a ring with a hole in the middle. The element in front of the user is the active element. To change the latter, the user must rotate the ring according to a predefined axis. This technique has been adopted by Gerber 2004 [119] but instead of displaying a complete circle before the user's hand, only a part of the menu is shown. The user selects an item by turning his wrist. Wesche and Droske 2000 [290], proposed a hybrid technique that uses a circular menu and pointer technique for selection. Each element is represented by an icon. The various icons are arranged in front of the user according to a portion of a circular arc and the selection is done using a virtual ray. A quick menu selection was introduced with the *Spin Menu* (Gerber and Bechmann 2005[118]). Items are arranged on a portion of a circle and controlled by rotating the wrist in the horizontal plane. Since 9-11 items can be displayed on a ring, hierarchical spin menus are suggested with crossed, concentric, or stacked layouts.

Glove-based menus: these allow more natural selection techniques using the fingers and hands. Typically, finger pinches are used to control a menu system. The

Tulip Menu (Bowman and Wingrave 2001[43]) is a drop-down menu based on the use of the fingers. The user's hand is equipped with a data glove and each finger is a menu item (Figure 2.13). The user can select a menu item by pinching the thumb to the appropriate finger. When the number of menu items is greater than four, it is possible to use both hands. The little finger is reserved for a "more options" item. When the menu was originally selected the first three item would appear on the first three fingers. Pinching the thumb to the little finger caused the next three items to appear in the non-dominant hand. This technique allows an accurate and fast selection. It is interesting because the user does not need to look at his finger to choose which one should be pinched. On the other hand, this technique is not suitable for a large menu.

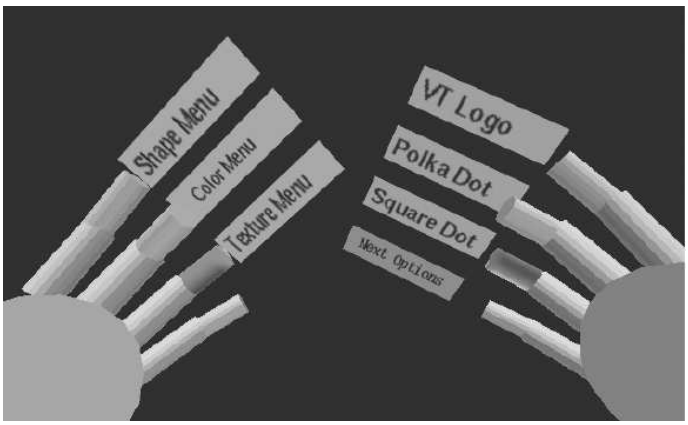


FIGURE 2.13: The tulip menu proposed by Bowman and Wingrave 2001[43].

Speech recognition enhanced menus: one of the problems resulting from the use of data gloves is that hands can be encumbered to use other tools (Hand 1997[137]). This was the motivation for the development of *hands-off interaction techniques* (Jacoby et*al.* 1994 [158]). The menu items are selected via speech recognition. Other menu solutions employ speech recognition as an alternative input in addition to graphical selection (multi-modal approach); among them the *3D Palette* by Billinghurst *et al.* 1997 [26].

Hand-held menus: allow menus to be controlled with only one hand, while the other hand is used to select items from it. Prominent examples are the interaction techniques developed by Mine *et al.* 1997 [199]. The *tear-off palette* contains miniature representations of available objects which can be selected and added to the virtual world by the user. Another example for two-handed direct menu selection is the *tool and object palette* which is based on tracked props (Szalavari and Gervautz 1997 [267]).

Workbench menus: these are very attractive for direct manipulation (Grosjean and Coquillart 2001 [128]). Typically, menus are used by means of a toolbox containing various 3D-icons. Interaction can be done with a stylus or data glove as

for example on the responsive workbench system introduced by Cutler et al. 1997 [83]. The C3 (Command and Control Cube) proposed by Grosjean and Coquillart [128] uses the *marking menu* concept. It proposes to place the menu at the cube rather than with a list, in order to accelerate access. The menu consists of 27 boxes divided into $3 * 3 * 3$ boxes. The purpose of this representation is to allow the user to select a command with a maximum three gesture to reach his goal. Each of the 26 boxes may contain a command, the 27th box, placed on the centre of the cube, is reserved for the special action to cancel the menu. The user's hand is represented by a small cursor always initialised at the centre of the cube. Thus, the user simply has to make a gesture in the direction of the desired item, then release the device button to perform the corresponding action. This technique gives excellent results in terms of speed selection and accuracy.

Body relative menus: these are attached to the user's body and thus take advantage of proprioception during operations. The *look-at-menu* (Mine *et al.* 1997 [199]) is an example. The menu can be attached to any object in the VE including the user. It is activated by the intersection of the user's direction of sight with a hot point representing a menu. To choose an item the user moves his/her head to simply look at the desired item to select it. Thus, head position is used instead of the traditional hand position. In 2007 Dachselt and Hbner [85] proposed a survey of graphical 3D menu solutions for VEs among others (e.g. augmented reality).

FIGURE 2.14: Immersive wall of the PREVISE platform [154].

2.6 Visual Display Configurations

Virtual Reality systems are based on different hardware configurations, mainly related to visual interfaces. These configurations can be classified into two categories:

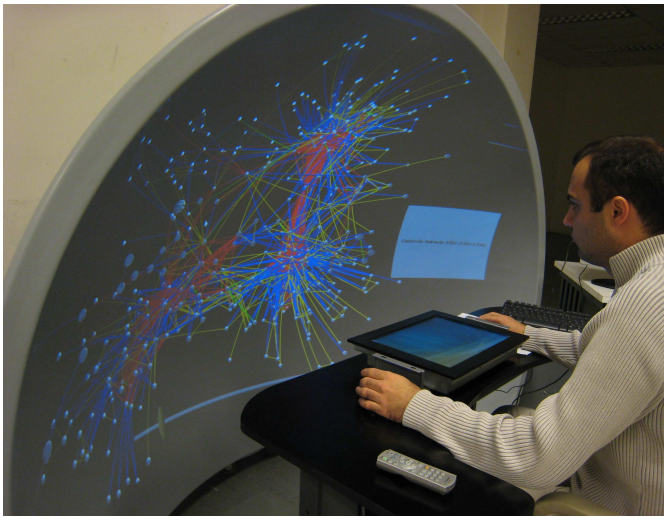


FIGURE 2.15: Example of immersive dome [126].

FIGURE 2.16: Example of immersive rooms [166].

- ***Immersive configurations*** (used in Fully-Immersive Virtual Environment (FIVE)) whose objective is to immerse the user in the VE (immersive virtual reality) through a stereoscopic display. Note that some configurations do not involve total recovery of the visual field of the user (wall, workbench);

- ***Non-immersive*** (used in Semi-Immersive Virtual Environment (SIVE)) configurations only allow the user to visualise virtual entities in a virtual world, through a display device such as a computer screen (Isdale 1993 [156]). Then, the user sees the world through a "window" represented by the screen. Note that this configuration does not necessarily imply the absence of Stereoscopic visualisation (objects appear behind or before the screen plane).

FIGURE 2.17: Example of workbench [262].

FIGURE 2.18: Example of the CAVE-like system [205].

2.6.1 Immersive configurations

Immersive configurations are based on one of the following visual interfaces.

- *Immersive walls*: these consist of a large flat screen on which images are projected. Visualisation, generally stereoscopic, is based on the use of polarising filters (linear or circular) and passive or active glasses for separating the images for the left eye of those for the right eye. The PREVISE platform of LISA is an immersive wall (Figure 2.14) which consists of a 2 x 2.5 m visual display with passive stereoscopic projection.

FIGURE 2.19: Example of head-mounted display [234].

- *Immersive domes*: these are hemispherical screens on which the image is projected (Figure. 2.15). The main advantage of this configuration is to propose a projection surface strengthening the sense of visual immersion through an important recovery of the user's visual field (Greffard et *al.* 2011 [126]).

- *Immersive rooms*: these are typically used to project virtual images on a large scale and allow a large number of users to simultaneously visualise the view. The main advantage of this configuration is its size which allows observation and exploration of large objects such as buildings (Figure 2.16).s

- *Workbenchs*: these have the form of a real workbench and allow the manipulation of objects in a natural and intuitive way (Paljic *et al.* 2002 [215], Grosjean *et al.* 2002 [127], Lecuyer *et al.* 2002[178], Steinicke *et al.* 2005 [262]). Virtual Workbenches are equipped with one or two stereoscopic displays and require the use of appropriate glasses (Figure 2.17).

- *Visiocubes*: the come in the form of a cubic enclosure having four to six large orthogonal screens. The stereoscopic display is back-projected (Figure 2.18). In the 6 screen configuration, the visual field of the user is completely covered and it no longer has any cue from the real world. The CAVE^{TM} (Computer Automatic Virtual Environment) Cruz-Neira et *al.* 1993 [82], the SAScube^{TM} (Nahon 2002 [205]), are two of the well-known visiocubes. Such visual displays are the most expensive ones and the most difficult to set-up and maintain.

- *Head-mounted displays (HMDs)*: these allow, via real time tracking of the actual position and orientation of the user's head, a full user immersion in the virtual world (Figure 2.19). The main advantage of HMDs is a better immersion of the user in the VE. However, its use is strictly individual and multiple headsets are

required for multi-user/or collaborative applications. Another drawback is that most HMDs have a quite small field of view (FOV).

It is important to also note that the use of HMDs presents ergonomic problems. Thus, the user mobility is often limited, given the wires connecting the visualisation device to the computer. On the other hand, its considerable weight and the low refresh rate may create discomfort and cybersickness.

2.6.2 Non-Immersive Configurations

In non-immersive (desk-top) configurations, the virtual environment is displayed on a PC screen or more recently on a 2D/3D TV screen. The size of the screen can therefore be quite large. These configurations are a relatively more affordable alternative that immersive configurations.

On a conventional screen, the virtual scene is displayed with quite better characteristics (resolution, light, etc.) than in an immersive (projected) configuration. Currently, the market of large size, low cost flat screen displays (LCD, OLED or plasma) allows us to consider non-immersive configurations, for which the displayed image occupies a larger portion of the user's visual field (a recent TV screen has a diagonal size of about 2m).

In this configuration, the VE is generally viewed through the screen in a non-colocalised configuration, i.e. with a spatial offset between the display space and the interaction space (Figure 2.20) or in a colocalised position (Figure 2.21).

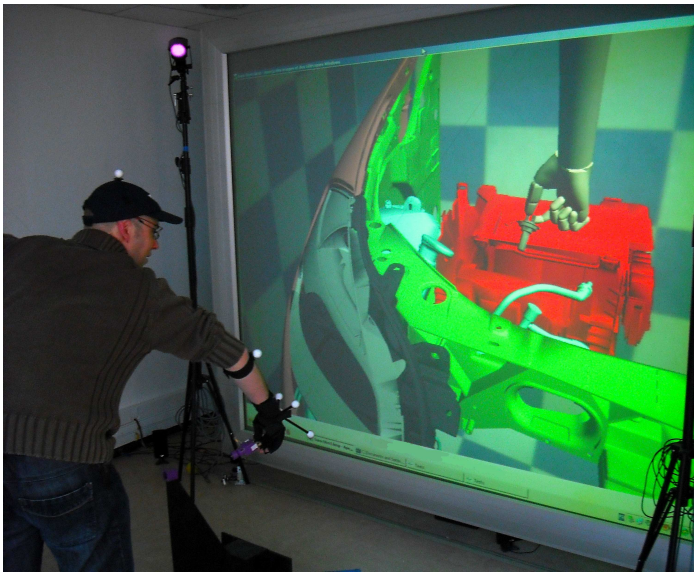


FIGURE 2.20: Illustration of a non-colocalised configuration [65].

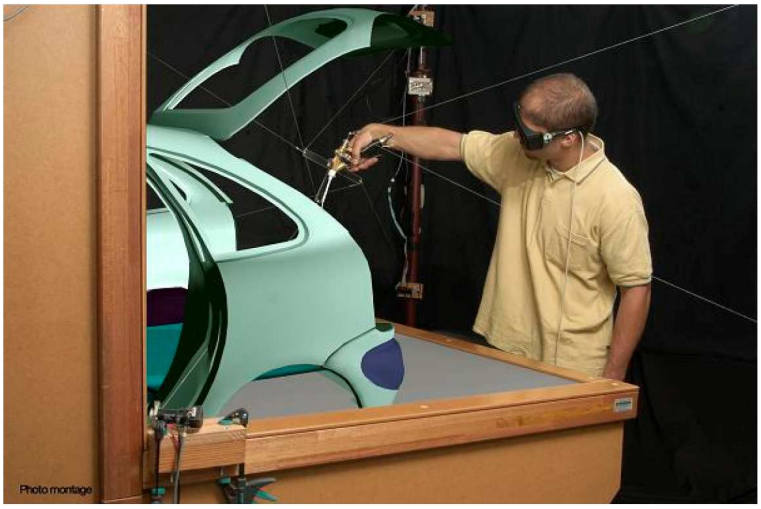


FIGURE 2.21: Illustration of a colocalised configuration [213].

Several configurations are available for non-immersive display systems such as :

- Screens directly connected to the computer: classic PC screens or more recently, large display (LCD or plasma);
- Laptop screens such as mobile phones screens or PDA (Personal Digital Assistants);
- Systems of video projection on various surfaces;
- Mixed solutions (Billinghurst *et al.* 2001 [27]).

The main interest of non-immersive configurations is that the display can be shared by several users (except laptop screens). In addition, systems using projectors or flat panel displays are independent of the user position and they are low-cost compared to HMDs. In addition, there are a variety of projection surfaces such as large screens or walls [232], semi-transparent mirrors (Rauterberg *et al.* 1997[231]), and surfaces (Piper *et al.* 2002[223]).

2.7 Conclusion

In this chapter, we proposed have a comparison between 2D, 3D and virtual reality techniques, and mentioned better suitability for fully immersive environments. Research work on the comparison between visual representations and interaction techniques have been presented. Then, we have presented and analysed 2D/3D visualisation and virtual reality configurations. In this context, we have giving an overview

of the existing interaction techniques that can be suited for Visual Data Miming (in the next Chapter we will present an overview of interaction techniques used in Visual Data Mining). Different classifications based on fundamental tasks such as navigation, selection and manipulation of virtual entities have also been analysed. Furthermore, we have presented a classification of visual displays, focusing on immersive and non-immersive ones. In the next chapter, we will propose a new classification of Visual Data Mining (VDM) techniques based on the use of 3D and virtual reality techniques.

3

Overview of Visual Data Mining in 3D and Virtual Reality

Contents

3.1 Introduction

At the output of the Data Mining (DM) process (post-processing), the decision-maker must evaluate the results and select what he finds interesting (Figure 3.1). Exploring and analysing the vast volume of knowledge extracted by the DM algorithms can be a complicated task. However, this task can be improved considerably with visual representations by taking advantage of human capabilities for perception and spatial cognition. Visual representations can allow rapid information recognition and show complex ideas with clarity and efficacy (Card *et al.*1999 [60]).

In everyday life, we interact with various information media which present us with facts and opinions based on knowledge extracted from data. It is common to communicate such facts and opinions in a virtual form, preferably interactive. For example, when watching weather forecast programmes on TV, the icons of a landscape with clouds, rain and sun, allow us to quickly build a picture about the weather forecast.

Such a picture is sufficient when we watch the weather forecast, but professional decision-making is a rather different situation. In professional situations, the decision-maker is overwhelmed by the DM algorithm results. Representing these results as static images limits the usefulness of their visualisation. This explains why the decision-maker needs to be able to interact with the data representation in order to find relevant knowledge. Interaction with the data representation can be exploited in two ways (de Oliveira and Levkowitz 2003 [89]):

- interaction with the DM algorithms results in facilitating and accelerating the analysis of studied data, the intermediate result, or produced knowledge.
- interaction with the DM algorithms.

Visual Data Mining (VDM), presented by Beilken and Spenke 1999 [23] as an interactive visual methodology "to help a user to get a feeling for the data, to detect interesting knowledge, and to gain a deep visual understanding of the data set", can facilitate knowledge discovery in data. The advantages of VDM is that the user is directly involved in the DM process and does not only interact with the data representation.

In 2D space, VDM has been studied extensively and a large number of visualisation techniques have been developed over the last decade to support the exploration of large data sets. Thus, including aesthetically appealing elements, such as 3D graphics and animation, increases the intuitiveness and memorability of visualisation. Also, it eases the perception of the human visual system (Spence 1990 [255], Brath *et al.* 2005 [47]). Although, there is still a debate concerning 2D vs 3D data visualisation (Shneiderman 2003 [245]), we believe that 3D and VR techniques have a better potential to assist the decision-maker in analytical tasks, and to deeply immerse the user in the data sets. In many cases, the user needs to explore data and/or knowledge from the

inside-out and not from the outside-in, as in 2D techniques (Nelson *et al.* 1999 [206]). This is only possible by using VR and Virtual Environment (VEs). VEs allow users to navigate continuously to new positions inside the data sets, and thereby obtain more information about the data. In KDD, VR has already been studied in different areas such as pre-processing (Nagel *et al.* 2008 [203], Ogi *et al.* 2009 [210]), classification (Einsfeld *et al.* 2006 [95]), and clustering (Ahmed *et al.* 2006 [4]). Although the benefits offered by VR compared to desk-top 2D and 3D still need to be proved, more and more researchers are investigating its use with VDM (Cai *et al.* 2007 [59]).

In this chapter, we firstly introduce research work on information visualisation regarding visualisation tool design and specially the graphic representation. In Section 2, we provide an overview of the current state of research concerning 3D visual representations. In Section 3, we present our motivation for interaction techniques in the context of KDD. In Section 4, we propose a new classification for VDM based on both 3D representations and interaction techniques. In addition, we survey representative works on the use of 3D and VR interaction techniques in the context of KDD. Finally, we present possible directions for future research.

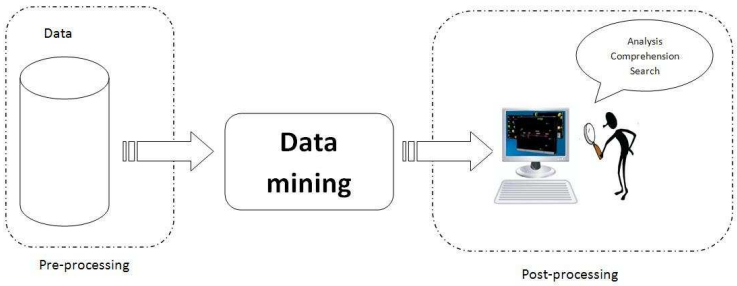


FIGURE 3.1: Illustration of the KDD process.

3.2 Visualisation

Visualisation is the display of information using graphic representations. A single picture can contain more information, and can be processed much more quickly than a page of words. This is because the human perceptual system can interpret images faster then text analysis. Pictures can also be independent of language, whereby a graph or a map may be understood by a group of people with no common language.

It is impressive to realise the number and types of visualised data that we encounter in our everyday activities. Some of these might include:

- train and subway maps;
- the instructions for changing a car headlight;

- a graph which may indicate the increase or decrease the number of unemployed in a country.

In each case, the visualisation provides an alternative to, or a supplement for, textual information. It is clear that visualisation provides a richer description of the information than the word-based counterpart.

3.2.1 Why is visualisation important ?

There are many reasons to explain why visualisation is important. The most obvious reason can be that sight is one of the human key senses for information understanding.

Figure 3.2 shows a diagram of an organisation that is difficult to describe verbally. However, the image can be easily comprehensible with only a brief examination (Ward *et al.* 2010 [286]). For instance, it is obvious that *marketing* has the most consultants and that *the driver* has the longest chain of command.

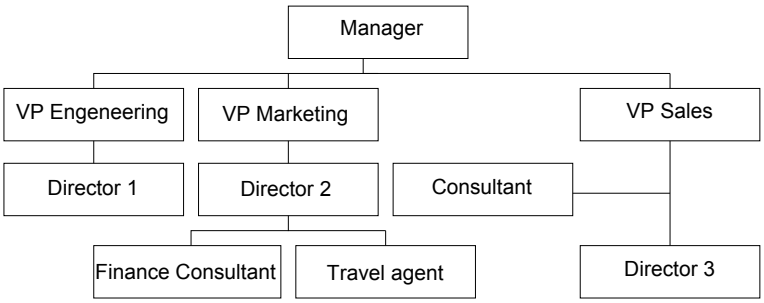


FIGURE 3.2: An organisation chart. A pattern requires at least one paragraph to describe it.

Example 3.2.1 In this example we highlight why visualisation is so important in decision making, and the role of human preferences and training.

Elting *et al.* 1999 [240] presented 34 clinicians with the preliminary results from hypothetical clinical trials of a generic conventional treatment compared with a generic investigation treatment; both treatments treating the same condition. Four different visualisation techniques are used to represent the two treatment results. The two treatments differed from one another and one of the treatments is better than the other. Clinicians seeing that difference should then decide to stop the trial.

Figure 3.3 shows the four presentations of the same data. In the upper left there is a simple table; in the upper right, pie charts; in the lower left, stacked bar charts; and in the lower right, a sequence of rectangles. In all representations, both the conventional and the investigation treatments are presented. The green colour shows that the drug induced a response and red that none occurred. The decision to stop varied significantly, depending on the presentation of the data. Correct decisions were

82% with icon display (lower right), 68% with tables, and 56% with pie charts or bar graphs. In actual clinical practice, up to 25% of the patients treated according to the data displayed as bar charts would have received inappropriate treatment. Clearly the choice of visualisation impacted the decision process. Elting *et al.* 1999 [240] noted that most (21/34) clinicians preferred the table, and that several did not like the icon display. This emphasises that it is not only the visualisation that is key in presenting data well, but that user preferences are strongly involved.

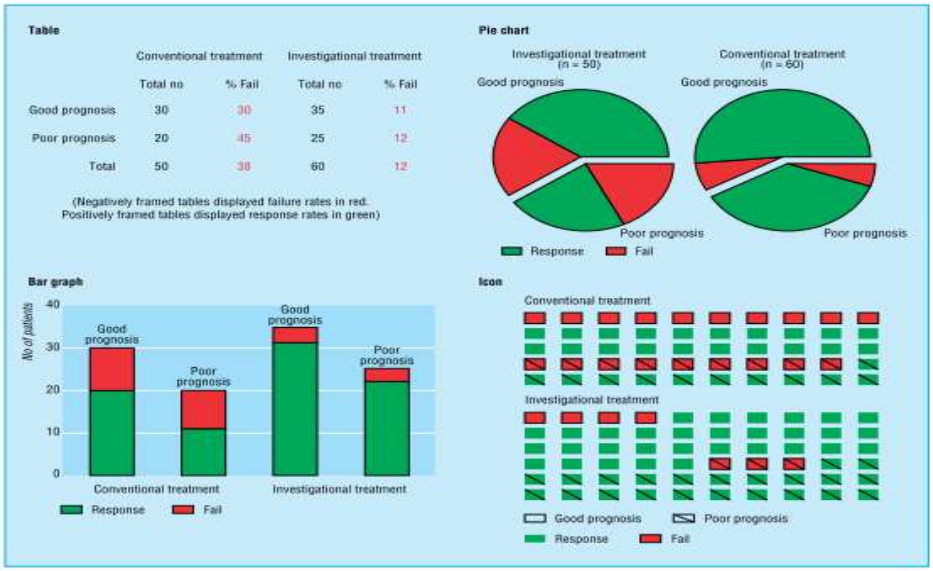


FIGURE 3.3: Four various visual representations of a hypothetical clinical trial. [240].

The high use of the web increase the amount of recorded data. The exploration and analysis of large marketing, financial, security, medical, and biological data sets produce results that need to be analysed. Given the increasing size of information available there is a growing need for tools and techniques to help make effective use of this information overload. Likewise, there is a growing need to find mechanisms for communicating information to people in an efficient and effective way, and to help educate them about processes and concepts that affect everyday life, from global warming to economic trends. Visualisation is paramount for these new knowledge discovery tools. Applications often use both static and interactive visualisation which are much more aesthetic and understandable to the user within applications to provide alternative views of the data and to help describe some structure, patterns, knowledge or anomaly in the data. In many domains, visualisation can be, and is becoming, an effective tool to assist user(s) in analysis and knowledge discovery.

What is the difference between scientific data visualisation and information visualisation ?

Both visualisations forms create visual representations from data that support user interaction with the aim of finding useful information in the data. In scientific visualisation, visual representations are typically constructed from measured or simulated data which represent objects or concepts of the physical world. Figure 3.4(a) shows an application that provides a VR interface to view the flow field around a space shuttle. In information visualisation, graphic models present abstract concepts and relationships that do not necessarily have a counterpart in the physical world, such as association rules (Agrawal *et al.* 1993 [2]), etc. For instance, Figure 3.4(b) shows a 3D tree representation to visualise data clusters.

(a) (b)

FIGURE 3.4: Scientific visualisation and information visualisation examples: (a): visualization of the flow field around a space shuttle (Laviola 2000 [177]) (b): GEOMIE (Ahmed *et al.* 2006 [4])information visualisation framework

3.2.2 The Visualisation Process

Visualisation is often part of a larger process, which may be exploratory data analysis, knowledge discovery, or visual analysis. To design a new visualisation tool, the designer should first begin with the analysis of the data to display. The data can come from different sources and have different structures (simple or complex). Then, the designer should consider the type of information that should be extracted from these data by the viewers. Visualisation can be used for different analysis tasks such as exploration (looking for interesting knowledge) or confirmation of hypotheses (based on prior beliefs).

To visualise data, the designer needs to define mapping from the data onto the display (Figure 3.5). There are many ways to realise this mapping and visualisation principles can provide mechanisms to translate date into more intuitive representation for users to perform their goals. This means that the data values are used to

define graphical objects, such as points, lines, and shapes, and their attributes like colour, position, size, and orientation. Thus, for example, we could map a number to a colour of a cube or a size of line to get a different way to view the same data.

Another, component of the visualisation process is interactive control of the viewing and mapping of variables. Whereas early visualisation was static, nowadays visualisation is a dynamic process. This means that the user controls all process stages, from data mapping to mapping and viewing. There is no effectiveness guarantee of a given visualisation. Different users, with different backgrounds, perceptual abilities, and preferences, will have different opinions on each visualisation. The user's task will also affect the usefulness and effectiveness of the visualisation. Each visualised datum change can have implications on the resulting visualisation. For this, it is essential to enable users to modify and interact with the visualisation until they achieve their goal, such as extracting an interesting piece of knowledge or confirming or denying hypothesis.

The visualisation process is traditionally described as a pipeline because is composed of a succession of stages which can be studied independently (Figure 3.5). This process starts with data to generate a visualisation. Among several visualisation pipelines, we selected the one proposed by Card *et al.* 1999 [60] that we think is the easiest and simplest representation of the visualisation process. We also noted two important points: user interaction can take place at any stage in the pipeline (nodes and link) and visualisation systems may have multiple views at the same time on the screen. Let us explain more precisely each transformations and stage.

- Data transformation: this is the starting stage of the visualisation process. It allows data to be transformed into something usable by the visualisation system. Data transformation deals with data issues such as data too large for visualisation. Large amounts of data may require ordering, selecting or filtering. For example, Williamson and Shneiderman 1992 [293] introduced *Dynamic queries* which allow users to formulate queries to select the data to be visualised by adjusting graphical widgets, such as sliders for quantitative variables and a check box for qualitative variables. The display is updated instantaneously.

- Rendering Mappings: once the data are ready, they can be represented. This requires data mapping into graphical objects by associating to each variable in the data to a graphic variable: shape, colour, and size, for example. The graphic objects can have from zero to three dimensions - that is point, line, areas and volumes. It is easy to simply develop a visualisation that conveys wrong information. Figure 3.6 shows an improper use of bar chart. By having the bars extend over each of the y-coordinate tick marks, there is an implication that the y-coordinate is involved, although no such association occurs. For example, the soybeans in the 5th column, cut across several y-values (INDIA, CHINA,

TURKEY) until reaching BRAZIL. A better representation is the one in Figure 3.7.

- View transformation: this is the final stage of the visualisation process. View transformation involves the presentation of graphical objects to the user. This includes interfacing with a computer graphics Application Programmer's Interface (API). The display view on the screen can be 3D or 2D. Many interaction techniques for view transformation are available allowing the user to change the view or the perspective onto the visual representation. The most known interaction techniques are:

  - *Viewpoint manipulation*: this is carried out by translation, rotation or zooming.
  - *Details on demand*: this consists of choosing an element in the representation and bringing up additional information about it (Shneiderman 1996 [244]).
  - *Focus/context* aimed to increase detailed description of certain parts of data (the point of interest, focus, etc), while the rest of the data is reduced in size, in order to provide guidance to the users. The best known *Focus/context* techniques are the techniques of distortion, such as the *Fish eye* proposed by Furnas 1986 [116]. In the technique of *bending backwards*, another variant of the *Focus/context* technique, the overview of different objects is not readable, but, miniatures of objects is an index to move directly to the information sought. However, there are other methods than the distortion of space. The viewing volume (Mroz and Hauser 2001 [201]) for example, proposes varying the opacity, (colour shades) and frequency to achieve *Focus/context* visualisation of 3D data.
  - *Brushing*: this means highlighting a selected subset of the data, but it can also be done to delete it from view, if the user wants to focus on other subsets of data.
  - *Multiple views*: this allows the user to have several views of the same representation on multiple windows. Interactive changes made in one visualisation are automatically reflected in the other visualisations. Connecting multiple visualisations through interactive *linking and brushing* provides more information than considering each visualisation independently.

Some techniques combine several types of transformation. For example, the semantic zoom is a view transformation which changes the data shown (data transformation). The more users zoom, the more the level of detail increase (Hascot and Beaudoin-Lafon 2001 [139]). From the visualisation process point of view, we can formalise the differences between the post-processing of association rule methods presented in Chapter 1 Section 1.5.3 as(Table 3.2.2):

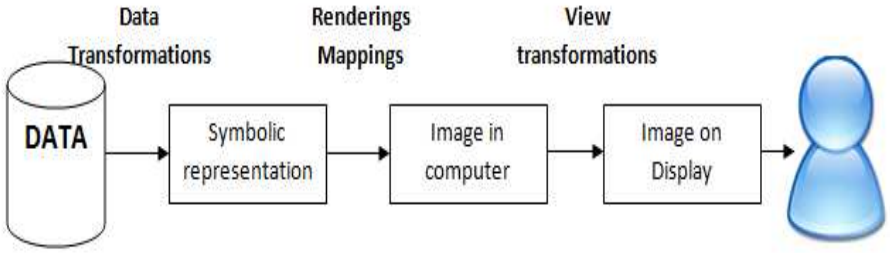


FIGURE 3.5: The visualisation process at a high level view [60].

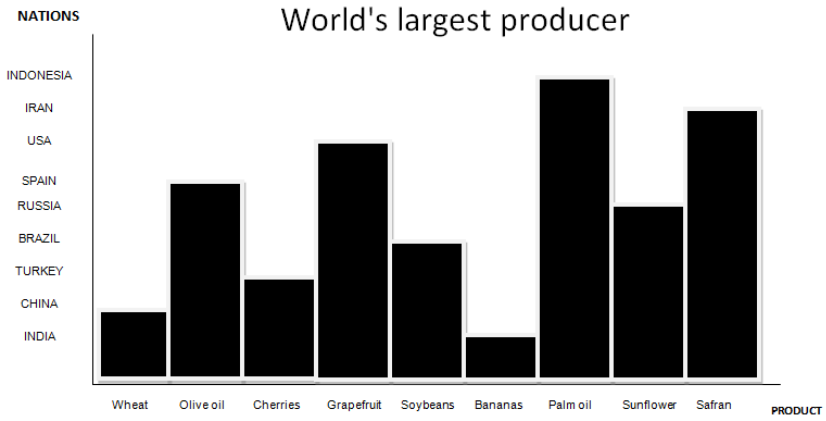


FIGURE 3.6: Poor use of a bar chart

- Association rule exploration methods (Query Languages and Rules Browser) only use data transformation;
- Visualisation of association rule mining result methods include Rendering Mappings and View transformations but are poor in Data transformation;
- Visualisation during association rule mining methods instantiate all transformations of the visualisation process.

As we will see later the post-processing of association rule methodology proposed in this thesis belongs to the 3rd category.

3.2.3 Semiology of graphics

Although many examples of different visualisation techniques have been proposed in order to determine the most effective encoding based on the variables to represent, we still lack a comprehensive language to describe our graphical creations. Robertson *et al.* 1991 [237] first suggested the creation of a formal model as a foundation for each

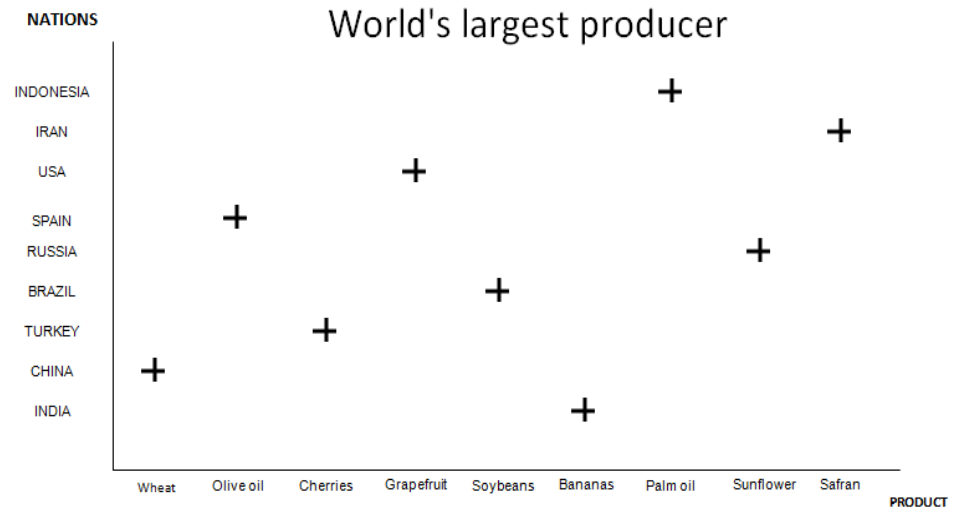


FIGURE 3.7: Better use of scatter plot

Methods	Data transformations	Rendering Mappings	View transformations
Query Languages	•		
Rule Browser	•		
Visualisation of association rule mining results		•	•
Visualisation during association rule mining	•	•	•

TABLE 3.1: Differences among the post-processing of association rules methods from the visualisation process point of view.

visualisation system. Among the authors who proposed a classification of encoding graphics, Cleveland and McGill 1984 [73] then Wilkinson 2005 [291] are the best know concerning static graphic (charts, scatter plots, etc.). The second flow of works comes from cartography, among them MacEachren 1995 [188] and more importantly Bertin 1967 [24].

In 1967, Bertin published his semiology of graphics (Bertin 1967 [24]). This was the first rigorous attempt to describe the link between data and visual elements. Although Bertin 1967 [24] draws his representations on paper, its principles are still the benchmark for today's representations.

Bertin presents the fundamentals of information encoding via graphic representations as a semiology, a science dealing with sign systems. His first key point is the strict separation of content (the information to encode) from the container (the properties of the graphic system).

Bertins theory states that graphical presentations use graphical marks, such as points, lines, and areas, to encode information via their positional and retinal properties (Bertin's graphical vocabulary is shown in Table 3.2.3).

Marks	Points, lines, and area
Positional	Two planar dimensions
Retinial	Size, value, texture, colour, orientation, and shape

TABLE 3.2: Bertin's graphical vocabulary [24].

Bertin's system is composed of seven visual graphic variables:

- Position (two planar dimensions);
- Six *retinal variables* (Figure 3.8): *size* (height, area, or number), *value* (saturation), *texture* (fitness or coarseness), *colour* (hue), *orientation* (angular displacement), and *shape*.

These graphic variables are combined with visual semantics for linking data attributes to visual elements.

Position plays a key role in visualisation because it is the visual information perceptually dominant in a visual representation (Bertin 1967 [24] , Card *et al.* 1999 [60], Wilkinson 2005 [291]). The other graphic variables called *retinal variables* (Table 3.8) are identified by experimental psychology and represent graphic variations designed for visual perception. There are called *retinal variables* because it is possible to perceive their variations without involving the muscles of the optical system unlike for the position. The *implantation* of *retinal variables* can be punctual, linear or zonal. Krygier and Wood 2005 [173] summarised the most effective use of Bertins retinal variables for showing qualitative or quantitative differences according to Bertin in Table 3.8.

As concerns size, this retinal variable denotes more surfaces than lengths. Therefore, graphical encoding based on surfaces are more relevant than the graphical encoding based on length. In some cases, the surface variation is reduced to a single length variation (in bar charts all rectangles have a side of the same length). To estimate the different possibilities of graphical encoding with the seven graphic variables, Bertin identified four possible attitudes for a person facing data (Bertin 1967 [24]):

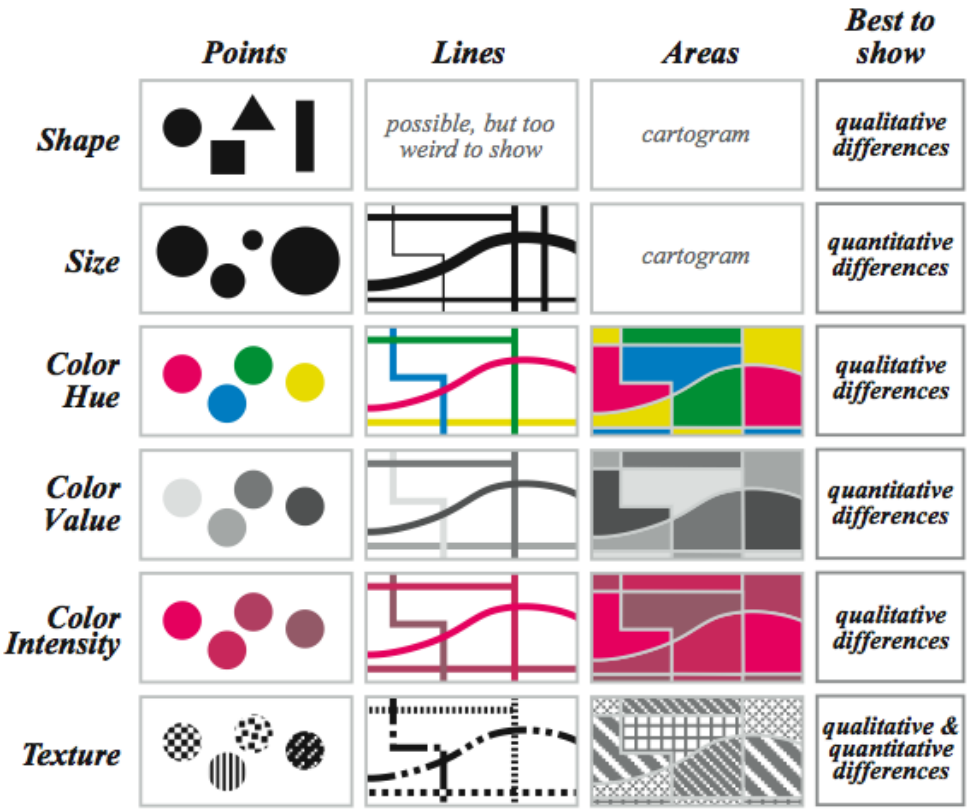


FIGURE 3.8: The most effective use of Bertins retinal variables [173].

- *Associative perception*: the user seeks to combine the different modalities of a dummy variable;
- *Selective perception*:the user seeks to distinguish the different modalities of a dummy variable;
- *Orderly perception*: the user seeks to perceive the order of ordinal variable modalities;
- *Quantitative perception*: the user seeks to perceive relationships among quantitative variable values.

Bertin summarises his principles of visualisation in the Table 3.2.3 which indicates which graphic variables are adapted to the data representation.

3.3 Visual Data Mining (VDM)

Historically, VDM has evolved from the fields of scientific visualisation and information visualisation (Section 3.2). Beilken *et al.* 1999 [23] presented the purpose of

Type of variable	Dummy		Ordinal	Quantitative
Perception	Associative	Selective	Orderly	Quantitative
Position	•	•	•	•
Size		•	•	•
Texture		•	•	
Colour	•	•	•	
Orientation	•	•		
Shape	•			

TABLE 3.3: Matching graphic variables and variables [24].

VDM as a way to "help a user to get a feeling for the data, to detect interesting knowledge, and to gain a deep visual understanding of the data set". Niggemann 2001 [208] looked at VDM as a visual representation of the data close to the mental model. We focus on the interactive exploration of data and knowledge that it is built on extensive visual computing (Gross 1994 [129]).

As humans understand information by forming a mental model which captures only the main information, in the same way, data visualisation, similar to the mental model, can reveal hidden information encoded in the data. In addition to the role of the visual data representation, Ankerst 2001 [10] explored the relation between visualisation and the KDD process. He defined VDM as "a step in the KDD process that utilises visualisation as a communication channel between the computer and the user to produce novel and interpreted patterns". He also explored three different approaches to VDM, two of which affect the final or intermediate visualisation results. The third approach involves the interactive manipulation of the visual representation of the data rather than the results of the KDD methods. The three definitions recognise that VDM relies heavily on human perception capabilities and the use of interactivity to manipulate data representations. The three definitions also emphasise the key importance of the following three aspects of VDM: visual representations; interaction processes; and KDD tasks.

In most of the existing KDD tools, VDM is only used during two particular steps of the KDD process: in the first step (pre-processing) VDM can play an important role since analysts need tools to view and create hypotheses about complex (i.e. very large and / or high-dimensional) original data sets. VDM tools, with interactive data representation and query resources, allow domain experts to quickly explore the data set (Ferreira-de-Oliveira and Levkowitz 2003 [89]). In the last step (post-processing) VDM can be used to view and to validate the final results that are mostly multiple and complex. Between these two steps, an automatic algorithm is used to perform the DM task. Some new methods have recently appeared which aim at

involving the user more significantly in the KDD process; they use visualisation and intensive interaction, with the ultimate goal of gaining insight into the KDD problem described by vast amounts of data or knowledge. In this context, VDM can turn the information overload into an opportunity by coupling the strengths of machines with that of humans. On the one hand, methods from KDD are the driving force of automatic analysis, while on the other side, human capabilities to perceive, relate and make conclusions turn VDM into a very promising research field. Nowadays, fast computers and sophisticated output devices can create meaningful visualisation and allow us not only to visualise data and concepts, but also to explore and interact with these data in real-time. Our goal is to look at VDM as an interactive process with the visual representation of data allowing KDD tasks to be performed. The transformation of data/knowledge into significant visualisation is not a trivial task. Very often, there are many different ways to represent data and it is unclear which representations, perceptions and interaction techniques needs to be applied. This proposed classification (Section 3.4) seeks to facilitate this task according to the data and the KDD goal to be achieved by reviewing representation and interaction techniques used in VDM. KDD tasks have different goals and diverse tasks need to be applied several times to achieve a desired result. Visual feedback has a role to play, since the decision-maker needs to analyse such intermediate results before making a decision.

3.3.1 3D Visual Representation for VDM

One of the problems that VDM must address is to find an effective representation of something that has no inherent form. In fact, it is crucial not only to determine which information to visualise but also to define an effective representation to convey the target information to the user. The design of a visualisation representation must address a number of different issues: what information should be presented? How this should be done? What level of abstraction to support? etc. For example, a user tries to find out interesting relations between variables in large databases. This information may be visualised as a graph (Prykeand and Beale 2005 [230]) or as an abstract representation based on a sphere and cone (Blanchard *et al.* 2007 [31]).

Many representations for VDM have been proposed. For instance, some visual representations are based on *abstract representations*, such as graphs (Ahmed *et al.* 2006 [4]), trees (Einsfeld *et al.* 2007 [96], Buntain 2008 [53]), and geometrical shapes (Ogi *et al.* 2009 [210], Nagel *et al.* 2008 [203], Meiguins *et al.* 2006 [192]) and others on *virtual worlds objects* (Baumgärtner *et al.* 2007 [17]).

The classification proposed in this section provides some initial insight into which techniques are oriented to certain data types, but does not assert that one visual representation is more suitable than others to explore a particular data set. Selecting a representation depends largely on the task being supported and is still a largely intuitive process.

Many researchers have attempted to construct a taxonomy for visualisation. Chi 2000 [71] used the Data State Model (Chi and Riedl 1998 [70]) to classify information visualisation techniques. This model is composed of 3 dimensions with categorical values: data stages (value, analytical abstraction, visualisation abstraction, and view), transformation operators (data transformation, visualisation transformation, and visual mapping transformation), and within stage operators (value stage, analytical stage, visualisation stage, and view stage). The separation of data states, visual transformations and operators provides flexibility in handling data/visual abstracts at different stages using operationally similar or functionally similar operators across different applications. This model shows how data change from one stage to another requiring one of the three types of data transformation operators. This state model helps implementers understand how to apply and implement information visualisation techniques. Tory and Moller 2004[277] present a high-level taxonomy for visualisation which classifies visualisation algorithms rather than data. Algorithms are categorised according to the assumption they make about the data being visualised. Their taxonomy is based on 2 dimensions: values of data (discrete or continuous) and how much the algorithm designer chooses display attributes (specialisation, timing, colour, and transparency). Teyseyre and Campo 2009 [273] presented an overview of 3D representations for visualising software, describing several major aspects like visual representations, interaction issues, evaluation methods, and development tools. They also performed a survey of some representative tools to support different tasks, i.e., software maintenance and comprehension, requirements validation, etc.

Not much work has attempted comparing visual representations. Wiss and Carr 1999 [294] performed a comparative study of 3 different 3D representations of information: a cam tree, an information cube, and an information landscape. In this study the authors chose to visualise a hierarchical file system. The task used to perform the study is based on the seven high-level information visualisation tasks as defined later by Shneiderman 2003 [245]: overview, zoom, filter, details-on-demand, relate, history and extract. The authors propose 3 tasks: *search*, based on Shneiderman's zoom task, *count*, based on Shneiderman's relate task, and *compare*, based on Shneiderman's overview task. This study showed that the possibility to get a good local and global overview is the one most important factors in supporting the types of tasks. Statistical analysis of the results show that the information cube performed worst for all tasks, and the *Information Landscape* performed best. The results indicate that the most influential factor was overview.

3.3.1.1 Abstract visual representations

3D representations are still abstract and require the user to learn certain conventions, because they do not look like what they refer to or they do not have a counterpart in the real-world. There are 3 kinds of abstract representations: graphs, trees, and geometrical shapes.

Graphs
A graph (Figure 3.9) is a network of nodes and arcs, where the nodes represent entities and the arcs represent relationships between entities. For a review on the state of the art in graph visualisation, see Herman *et al.*2000 [142] .

Originally, graph visualisation was used in 2D space to represent components around simple boxes and lines. However, several authors think that larger graph structures can be viewed in 3D (Parker *et al.* 1998 [216]).

A technique based on the hyper system (Hendley *et al.* 1999 [141]) for force-based visualisation can be used to create a graph representation. The visualisation consists of nodes and links whose properties are given by the parameters of the data. Data elements affect parameters such as node size and colour, link strength and elasticity. The dynamic graphs algorithm enables the self-organisation of nodes in the visualisation area by the use of a force system in order to find a steady state, and determine the position of the nodes. For example, Beale *et al.* 2007 [20] proposed a Haiku system (Figure 3.9 (b)) which provides an abstract 3D perspective of clustering algorithm results based on the hyper system. One of the characteristics of this system is that the user can choose which parameters are used to create the distance metrics (distance between two nodes), and which ones affect the other characteristics of the visualisation (node size, link elasticity, etc.). Using the hyper system allows related things (belonging to the same cluster) to be near to each other, and unrelated things to be far away.

Trees
3D trees (Figure 3.10) is a visualisation technique based on the hierarchical organisation of data. A tree can represent many entities and the relationships among them. In general, the visualisation of hierarchical information structures is an important topic in the information visualisation community (van-Ham and van-Wijk 2002 [284]). Be cause trees are generally easy to layout and interpret (Card *et al.*1999 [60]), this approach finds many applications in classification visualisation (Buntain 2008 [53]). 3D trees were designed to display a larger number of entities than in 2D representations, in a comprehensible form (Wang *et al.* 2006 [285]). Various methods have been developed for this purpose, among which, space-filling techniques and node-link techniques.

Space-filling techniques (Van-ham and Van-wijk 2002 [284], Wang *et al.* 2006 [285]) based upon 2D tree-maps visualisation proposed by Johnson and Shneiderman 1991 [161] have been successful for visualising trees that have attribute values at the node level. Space-filling techniques are particularly useful when users care mostly about nodes and their attributes but do not need to focus on the topology of the tree, or consider that the topology of the tree is trivial (e.g 2 or 3 levels). The users of space-filling techniques also require training because of the unfamiliar layout (Plaisant *et al.* 2002 [224]).

Node-link techniques, on the other hand, have long been frowned upon in the

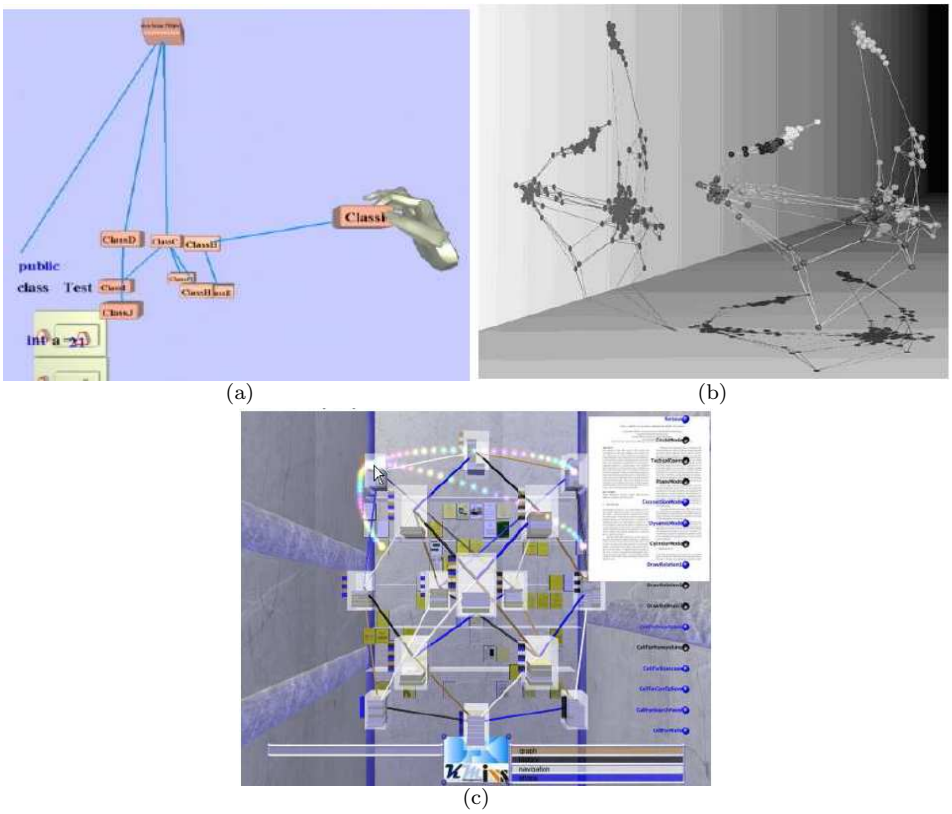


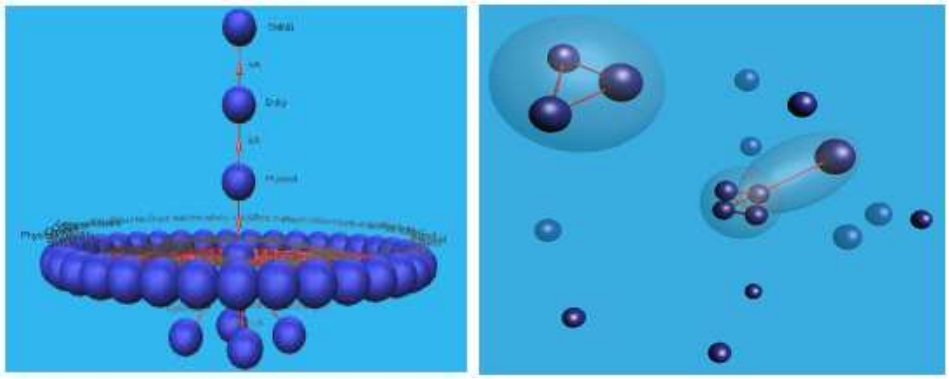
FIGURE 3.9: An example of graph representations: (a) Ougi[214], (b) Association rules: Haiku [230], (c) DocuWorld [95]

information visualisation community because they typically make inefficient use of screen space. Even trees of a hundred nodes often need multiple screens to be completely displayed, or require scrolling since only part of the tree is visible at a given time. A well-known node-link representation in cone trees was introduced by Robertson *et al.* 1991 [237] for visualising large hierarchical structures in a more intuitive way. 3D trees may be displayed vertically (Cone Tree) or horizontally (Cam Tree).

Buntain 2008 [53] used 3D trees for ontology classification visualisation (Figure 3.10 (a)). Each leaf represents a unique concept in the ontology, and the transparency and size of each leaf is governed by the number of documents associated with the given concept. A molecule is constructed by clustering together spheres that share common documents, and surrounds the leaves with a semi transparent shell (Figure 3.10 (b)).

Geometrical shapes

In this technique, 3D objects with certain attributes are used to represent data and knowledge. The 3D scatter-plot visualisation technique (Nagel *et al.* 2001 [204]) is one of the most common representations based on geometric shapes (Figure 3.11). The main innovation compared to 2D visualisation techniques is the use of volume



(a) (b)

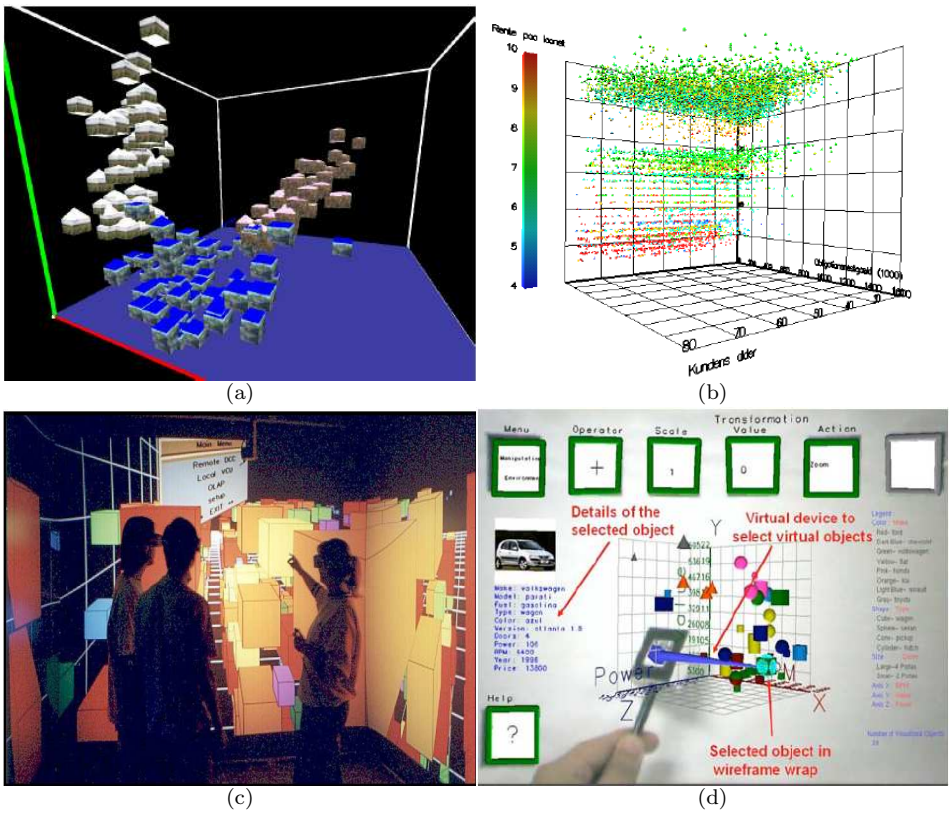
FIGURE 3.10: An example of tree representing ontology classification : SUMO [53]

rendering which is a conventional technique used in scientific visualisation. 3D rendering techniques use voxels (instead of pixels in 2D) to present a certain density of the data. 3D scatter-plot have been adapted by Becker 1997 [21], making the opacity of each voxel a function of point density. Using scatter-plots is intuitive since each datum is faithfully displayed. Scatter-plots have been used successfully for detecting relationships in two dimensions (Bukauskas and Bhlen 2001[52], Eidenberger 2004 [94]). This technique hit limitations if the dataset is large, noisy, or if it contains multiple structures. With large amounts of data, the amount of displayed objects makes it difficult to detect any structure at all.

3.3.1.2 Virtual worlds

Trying to find easily-understandable data representations, several researchers proposed the use of real-world metaphors. This technique uses elements of the real-world to provide insights about data. For example, some of these techniques are based on a city abstraction (Figure 3.12). The virtual worlds (sometimes called cyber-spaces) for VDM are generally based either on the information galaxy (Krohn 1996 [172]) or the information landscape metaphor (Robertson *et al.* 1998 [236]). The difference between the two metaphors is that in the information landscape, the elevation of objects is not used to represent information (objects are placed on a horizontal floor). The specificity of virtual worlds is that they provide the user with some real world representations.

Trying to find easily understandable representations of data, several researchers proposed using real-world metaphors. These techniques use elements of the world to provide insight about data. For example, some of these techniques are based on a city abstraction (Figure 3.12). In Imsovision – IMmersive SOftware VISualisatION (Maletic *et al.* 2001 [190]), the platform size is proportional to the size of the class (i.e., number of methods and attributes). The attributes of a class are viewed as spheres and number functions viewed as columns. This type of natural representation reduces the cognitive overhead of the visualisation.



(a) (b) (c) (d)

FIGURE 3.11: Different 3D scatter plots representations: (a) VRMiner [14], (b) 3DVDM [203], (c) DIVE-ON [8], (d) Visualisation with augmented reality[192]

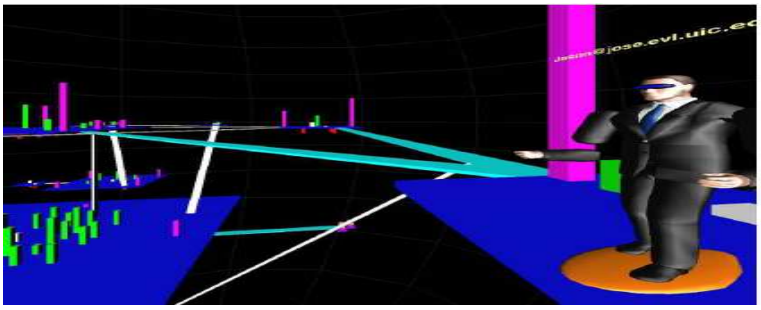


FIGURE 3.12: Example of virtual worlds representation: Imsovision [190].

3.3.2 Interaction for VDM

Data Mining (DM) usually deals with massive data sets and most visual techniques propose mapping each data item into a corresponding graphical element (pixel, line, icon, etc.). The implication is that they do not scale well when handling millions of

data items. This made clear the need to interact with the visual representations to reduce visual clutter and object overlap. Interaction techniques can provide the user with mechanisms for handling complexity in large data sets.

VDM is not a completely human guided process. DM algorithms analyse a data set searching for useful information and statistically valid patterns. The degree of automation of data mining process varies, but still the algorithm, not the user, which is the one that is looking for patterns. In this context, Ferreira-de-Oliveira and Levkowitz 2003 [89] denoted that VDM should have a larger role than traditional applications of visualisation techniques. This creates the potential of greatly increasing user participation in the DM process as a whole, as well as the end user's overall understanding of the process. To make it feasible will certainly require a greater interaction between the information visualisation and DM communities. This allows increasing recognition within the information visualisation that interaction and inquiry are inextricable and that it is through the interactive manipulation of a visual interface that knowledge is constructed, tested, refined and shared.

Interaction techniques can empower the user's perception of information when visually exploring a data set (Hibbard *et al.* 1995 [143]). The ability to interact with visual representations can greatly reduce the drawbacks of visualisation techniques, particularly those related to visual clutter and object overlap, providing the user with mechanisms for handling complexity in large data sets. Pikea *et al.* 2009 [292] explored the relationship between interaction and cognition. They consider that the central percept of VDM is that the development of human insight is aided by interaction with a visual interface. As VDM is concerned with the relationship between visual displays and human cognition, merely developing only novel visual metaphors is rarely sufficient to make new discoveries provide confirm or negate prior beliefs.

We can distinguish three different interaction categories: exploration, manipulation and human-centred approaches.

3.3.2.1 Visual exploration

Visual exploration techniques are designed to take advantage of the considerable visual capabilities of human beings, especially when users try to analyse tens or even hundreds of graphic variables in a particular investigation. Visual exploration allows the discovery of data trends, correlations and clusters, to take place quickly, and can support users in formulating hypotheses about the data. It is essential in some situations to allow the user to simply look at the visual representation in a passive sense. This may mean moving around the view point in order to reveal structure in the data that may be otherwise masked and overlooked . In this way, exploration provides the means to view information from different perspectives to avoid occlusion and to see object details. It can be very useful to have the ability to move the image to resolve any perceptual ambiguities that exist in a static representation when a large amount of information is displayed at once. The absence of certain visual cues (when viewing

a static image) can mask important results (Kalawsky and Simpkin 2006 [165]).

Navigation is often the primary task in 3D worlds and refers to the activity of moving through the scene. The task of navigation presents challenges such as supporting spatial awareness and providing efficient and comfortable movement among distant locations. Some systems enable users to navigate without constraint through the information space (Nagel *et al.* 2008 [203], Einsfeld *et al.* 2006 [95], Azzag *et al.* 2005 [14]). Other systems restrict movement in order to reduce possible user disorientation (Ahmed *et al.* 2006 [4]). As an illustration, in VRMiner (Azzag *et al.* 2005 [14]) a six-degree freedom sensor is fixed to the user's hand (Figure 3.13) allowing him/her to easily define a virtual camera in 3D space. For example, when the user moves his hand in the direction of the object, he/she may zoom in or out. The 3DVDM system (Nagel *et al.* 2008 [203]) allows the user to fly around and within the visualised scatter-plot. The navigation is controlled by the direction of a "wanda" device tracked with 6 degrees of freedom. Dissimilarly, in GEOMI (Ahmed *et al.* 2006 [4]), the user can only rotate the representation along the X and Y axes but not along the Z axis.

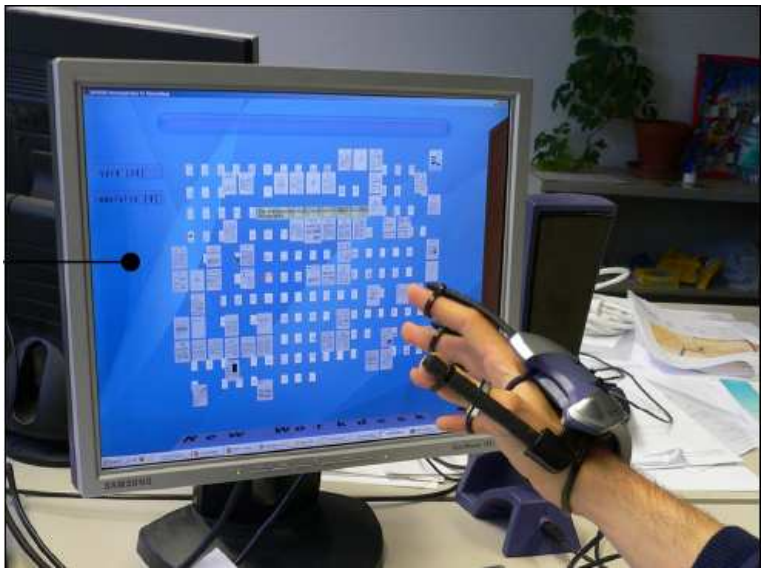


FIGURE 3.13: Illustration of a navigation technique based on the use of a data glove [17].

In visual exploration, the user can also manipulate the objects in the scene. In order to do this, interaction techniques provide means to select and zoom-in and zoom-out to change the scale of the representation. Beale *et al.* 2007 [20] demonstrated that using a system which supports the free exploration and manipulation of information delivers increased knowledge even from a well-known dataset. Many systems provide a virtual hand or a virtual pointer (Einsfeld *et al.* 2007 [96]), a typical approach used in VE, which is considered as being intuitive as it simulates real-world interaction (Bowman *et al.* 2001[41]).

- Select: this technique provides users with the ability to mark interesting data items in order to keep track of them when too many data items are visible, or when the perspective changes. In these two cases, it is difficult for users to follow interesting items. By making items visually distinctive, users can easily keep track of them even in large data sets and/or with changed perspectives.

- Zoom: by zooming, users can simply change the scale of a representation so that they can see an overview (context) of a larger data set (using zoom-out) or the detailed view (focus) of a smaller data set (using zoom-in). The essential purpose is to allow hidden characteristics of data to be seen. A key point here is that the representation is not fundamentally altered during zooming. Details simply come into focus more clearly or disappear from view.

Visual exploration (as we can see in Section.7) can be used in the pre-processing of the KDD process to identify interesting data (Nagel *et al.* 2008 [203]), and in post-processing to validate DM algorithm results (Azzag *et al.* 2005 [14]). For example, in VRMiner(Azzag *et al.* 2005 [14]) and in ArVis (Blanchard *et al.* 2007 [31]), the user can select an object to obtain information about it.

3.3.2.2 Visual manipulation

In KDD, the user is essentially faced with a mass of data that he/she is trying to make sense of. He/she should look for something *interesting*. However, *interest* is an essentially human construct, a perspective of relationships among data that is influenced by tasks, personal preferences, and past experience. For this reason, the search for knowledge should not only be left to computers; the user has to guide it depending upon what he/she is looking for, and hence which area to focus computing power on. Manipulation techniques provide users with different perspectives of the visualised data by changing the representation.

One of these techniques is the capability of changing the attributes presented in the representation. For example, in the system shown by Ogi *et al.* 2009 [210], the user can change the combination of presented data. Other systems have interaction techniques that allow users to move data items more freely in order to make the arrangement more suitable for their particular mental model (Einsfeld *et al.* 2006 [95]).

Filter interaction techniques enable users to change the set of data items being presented on some specific conditions. In this type of interaction, the user specifies a range or condition, so that only data meeting those criteria are presented. Data outside the range or not satisfying the conditions are hidden from the display or shown differently; even so, the actual data usually remain unchanged so that whenever users reset the criteria, the hidden or differently-illustrated data can be recovered. The user is not changing data perspectives, just specifying conditions within which data are shown. ArVis (Blanchard *et al.* 2007 [31]), allows the user to look for a rule with a particular item in it. To do this, the user can search for it in a menu which lists all the rule items and allows the wanted object to be shown.

3.3.2.3 Human-centred approach

In most existing KDD tools, interaction can be used in two different ways: exploration and manipulation. Some new methods have recently appeared (Baumgärtner *et al.* 2007 [17], Poulet and Do 2008 [225]), tried to involve the user in the DM process more significantly and used visualisation and interaction more intensively. In this task, the user manipulates the DM algorithm and not only the graphical representation. The user sends commands to the algorithm in order to manipulate the data to be extracted. We are speaking here about local knowledge discovery. This technique allows the user to focus on interesting knowledge from his/her point of view, in order to make the DM tool more generically useful to the user. It is also necessary for the user to either change the view point or manipulate a given parameter of the knowledge discovery algorithm and observe its effect. There must therefore be some way in which the user can indicate what it is considered interesting and what is not, and to do this the KDD tool needs to be dynamic and versatile (Ceglar *et al.* 2003[63]). The human-centred process should be iterative since it is repeated until the desired results are obtained. From a human interaction perspective, a human-centred approach closes the loop between the user and the DM algorithm in a way that allows them to respond to results as they occur by interactively manipulating the input parameters (Figure 3.14). With the purpose of involving more intensively the user in the KDD process, this approach has the following advantages (Poulet and Do 2008 [225]).

- the quality of the results is improved by the use of human-knowledge recognition capabilities;
- using domain knowledge during the whole precess (and not only in the interpretation of the results) allows guided searching for knowledge;
- the confidence in the results is improved as the DM process gives more comprehensible results.

In Arvis (Blanchard *et al.* 2007 [31]), the user can navigate among the subsets of rules via a menu providing neighbourhood relations. By applying a neighbourhood relation to a rule, the mining algorithm extracts a new subset of rules. The previous subset is replaced by the new subset in the visualisation area.

3.4 A New Classification for VDM

In this section, we present a new classification of VDM tools based on 3 dimensions: visual representations, interaction techniques, and KDD tasks. Table 3.4 presents the different modalities of each of the three dimensions. The proposed taxonomy takes into account both visual representation and interaction techniques. In addition, many visualisation design taxonomies include only a small subset of techniques (e.g., locomotion Arns 2002 [11]). Currently, visualisation tools have to provide not only effective visual representations but also effective interaction metaphors to facilitate exploration and help users achieve insight. Having a good 3D representation without

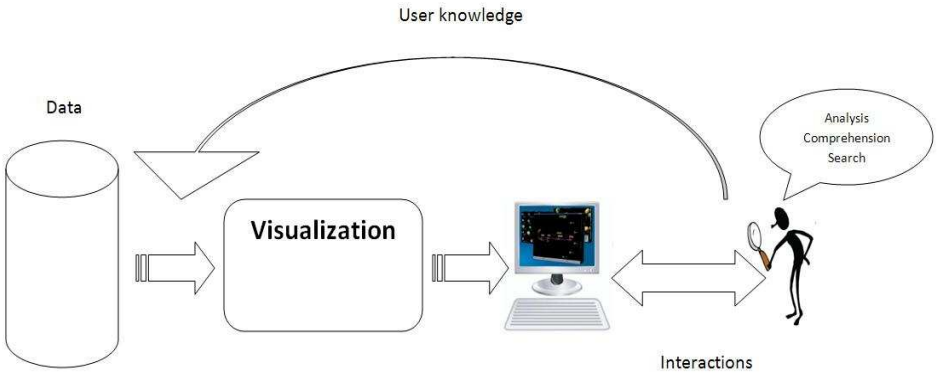


FIGURE 3.14: Illustration of the human-centred approach.

a good interaction technique does not mean having a good tool. Existing metaphors for visualisation and interaction can be classified under the new system, enabling designers to more easily compare metaphors to see exactly how they are different and similar to each other. This classification looks at some representative tools for doing different KDD tasks, e.g., pre-processing and post-processing (classification, clustering and association rules). Different tables summarise the main characteristics of the reported VDM tools with regard to visual representations and interaction techniques. Other relevant information such as interaction actions (navigation, selection and manipulation, and system control), input-output devices (CAVEs, mice, hand trackers, etc.) presentation (3D representation or VR representation) and year of creation is also reported.

Dimension	**Modalities**
Visual representation	Graphs, 3D trees, geometrical shapes, virtual worlds
Interaction techniques	Visual exploration, visual manipulation, human-centred
KDD tasks	Pre-processing, classification, clustering, association rules

TABLE 3.4: Dimension modalities

3.4.1 Pre-processing

Pre-processing (in VDM) is the task of data visualisation before the DM algorithm is used. It is generally required as a starting point of KDD projects so that analysts may identify interesting and previously unknown data by the interactive exploration of graphical representations of a data set without heavy dependence on preconceived assumptions and models. The basic visualisation technique used for data pre-processing

System	Visual Representation	Interaction techniques	Interaction actions			Input-Output devices	3D/VR	year
			Navigation	Selection and Manipulation	System control			
Ogi *et al.* 2009[210]	Geometric shape	Visual manipulation	-	-	Graphical menu	CAVE	VR	2009
3DVDM Nagel *et al.* 2008 [203] (Figure 3.11(b))	Geometric shape	Visual exploration	Manual view point manipulation + thought wizard metaphor	-	-	Three-button 'wand' + CAVE	VR	2008
Nested circles Wang *et al.* 2006 [285]	Visual exploration and visual manipulation	Tree	-	Object selection + virtual pointer	Graphic menus	Mouse + 2D screen	3D	2006
Visualisation with augmented reality Meiguins *et al.* 2006 [192] (Figure 3.11(d))	Geometric shape	Visual manipulation	-	Object selection + virtual hand	Graphical menus	Hand tracker	VR	2006
Dive-On Ammoura *et al.* 2001[8] (Figure 3.11(c))	Geometric shape	Visual exploration and manipulation	Physical movement + steering + target-based travel	Object selection	Graphical menus	Hand + head tracker + CAVE	VR	2001

TABLE 3.5: 3D VDM tool summary for pre-processing KDD task.

is the 3D scatter-plot method, where 3D objects with attributes are used as markers. The main principle behind the design of traditional VDM techniques, such as The Grand Tour (Asimov 1985 [13]), the parallel coordinate (Inselberg and Dimsdale 1990 [155]), etc., is that they are viewed from the outside-in. In contrast to this, VR lets users explore the data from inside-out by allowing users to navigate continuously to new positions inside the VE in order to obtain more information about the data. Nelson *et al.*1999 [206] demonstrated through comparisons between 2D and VR versions of the VDM tool XGobi that the VR version of XGobi performed better. In the system proposed by Ogi *et al.* 2009 [210], the user can see several data set representations integrated in the same space. The user can switch the visible condition of each data set. This system could be used to represent the relationships among several data sets in 3D space, but it does not allows the user to navigate through the data set and interact with it. The user can only change the visual mapping of the data set. However, the main advantage of this system is that the data can be presented with a high degree of accuracy using high-definition stereoscopic display that can be beneficial especially when visualising a large amount of data. This system has been applied to the visualisation and analysis of earthquake data. Using the 3rd dimension has allowed the visualisation of both the overall distribution of the hypocentre data

and the individual location on any earthquake, which is not possible with the conventional 2D display. The Figure 3.15 shows hypocentre data recorded over 3 years.

FIGURE 3.15: Visualisation of earthquakes data using a 4K stereo projection system [210].

The system allows the visualisation of several databases at the same time e.g. map data, terrain data, basement depth, etc and the user can switch the visible condition of each datum in the VE. For example, the user can change the visualisation data from the combination of hypocentre data and basement depth data to the combination of hypocentre data and terrain data. Thus, the system can show the relationships between, only, any two data sets among the others.

As a result of using VR, the 3DVDM system (Nagel *et al.* 2008 [203]) is capable of providing real-time user response and navigation as well as showing dynamic visualisation of large amounts of data. Nagel *et al.*2008 [203] demonstrated that the 3DVDM visualisation system allows faster detection of non-linear relationships and substructures in data than traditional methods of data analysis. An alternative proposal is available with DIVE-ON (Data mining in an Immersed Visual Environment Over a Network) system, proposed by Ammoura *et al.* 2001[8]. The main idea of DIVE-ON is visualising and interacting with data from distributed data warehouses in an immersed VE. The user can interact with such sources by walking or flying towards them. He/she can also can pop up a menu, scroll through it and execute all environment, remote, and local functions. Thereby, DIVE-ON makes intelligent use of the natural human capability of interacting with spatial objects and offers considerable navigation possibilities e.g. walking, flying, transporting and climbing.

Inspired by *Treemap* Wang *et al.* 2006 [285] presented a novel space-filling approach for tree visualisation of file systems (Figure 3.16). This system provides a

good overview for a large hierarchical data set and uses nested circles to make it easier to see groupings and structural relationships. By clicking on an item (a circle), the user can see the associated sub-items represented by the nested circles in a new view. The system provides the user with a control panel allowing him/her to filter files by types; by clicking on one file type, the other files types are filtered out. A zoom-in/zoom-out function allows the user to see folder or file characteristics such as name, size, and date. A user-feedback system means that user interaction techniques are friendly and easy to use.

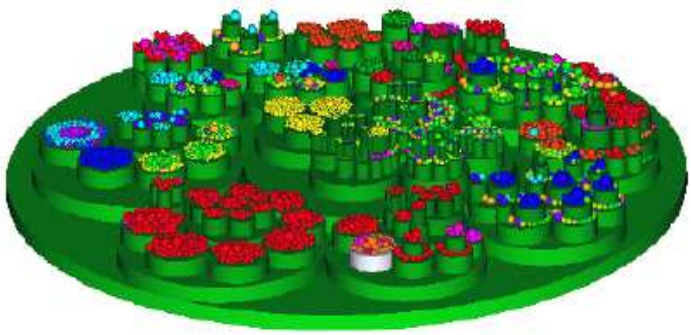


FIGURE 3.16: Representation of a file system with 3D-nested cylinders and spheres [285].

Meiguins *et al.* 2006 [192] presented a tool for multidimensional VDM visualisation in an augmented-reality environment where the user may visualise and manipulate information in real time VE without the use of devices such as a keyboard or mousse and interact simultaneously with other users in order to make a decision related to the task being analysed. This tool uses a 3D scatter-plot to visualise the objects. Each visualised object has specific characteristics of position (x, y and z axes), colour, shape, and size that directly represent data item values. The main advantages of these tools is that they provide users with a dynamic menu which is displayed in an empty area when the user wants to execute certain actions. The tool also allows users to perform many manipulation interaction tasks such as real-time filter attributes, semantic zoom, rotation and translation of objects in the visualisation area. A detailed comparison of these techniques is presented in Table 3.5.

3.4.2 Post-processing

Post-processing is the final step of the KDD process. Upon receiving the output of the DM algorithm, the decision-maker must evaluate and select the interesting part of the results.

3.4.2.1 Clustering

Clustering is used for finding groups of items that are similar. Given a set of data items, this set can be partitioned into a set of classes, so that items with similar characteristics are grouped together.

The GEOMI system proposed by Ahmed *et al.* 2006 [4] is a visual analysis tool for the visualisation of clustered graphs or trees. The system implements block model methods to associate each group of nodes to corresponding cluster. Two nodes are in the same cluster if they have the same neighbour set. This tool allows immersive navigation in the data using 3D head gestures instead of the classical mouse input. The system only allows the user visual exploration. Users can walk into the network, move closer to nodes or clusters by simply aiming in their direction. Nodding or tilting the head rotates the entire graph along the X and Y axes respectively, which provides users with intuitive interaction.

The objective of *@VSIOR* (Baumgartner *et al.* 2007 [17]), which is a human-centred approach, is to create a system for interaction with document, meta-data, and semantic relations. Human capabilities in this context are spatial memory and the fast visual processing of attributes and patterns. Artificial intelligence techniques assist the user, e.g. in searching for documents and calculating document similarities. Similarly, VRMiner (Azzag *et al.* 2005 [14]) uses stereoscopic and intuitive navigation; these allow the user to easily select the interesting view point. VRMiner users have found that using this tool helps them solve 3 major problems: detecting correlation between data dimensions, checking the quality of discovered clusters, and presenting the data to a panel of experts. In this context, the stereoscopic display plays a crucial role in addition to the intuitive navigation which allows the user to easily select the interesting view point. A detailed comparison of these techniques is presented in Table 3.6.

System	**Visual Representation**	**Interaction techniques**	**Interaction actions**			**Input-Output devices**	**3D/ VR**	**year**
			Navigation	**Selection and Manipulation**	**System control**			
@VISOR Baumgartner *et al.* 2007 [17] (Figure 3.13)	Graph	Human- centred	-	Object selection + object positioning+ virtual hand	Graphical menu + gestural interaction	Tablet PC(2D) + Data glove + stereoscopic	VR	2007
GEOMI Ahmed *et al.* 2006 [4]	Graph + tree	Visual exploration	Steering	-	Gestural interaction	Head tracker + stereoscopic	VR	2006
VRMiner Azzag *et al.* 2005 [14] (Figure 3.11(a))	Abstract geometrical shape	Visual exploration	Manual view point manipulation	Object selection	Gestural interaction	Data glove + stereoscopic	VR	2005

TABLE 3.6: 3D VDM tool summary for clustering KDD task

3.4.2.2 Classification

Given a set of pre-defined categorical classes, how can one determine which of these classes a specific data item belongs to? In SUMO (Figure 3.10), a tool for document-class visualisation is proposed (Buntain 2008 [53]). The classes and relations among them can be presented to the user in a graphic form to facilitate understanding of the knowledge domain. This view can then be mapped onto the document space where shapes, sizes, and locations are governed by the sizes, overlaps, and other properties of the document classes. This view provides a clear picture of the relations among the resulting documents. Additionally, the user can manipulate the view to show only those documents that appear in a list of a results from a query. Furthermore, if the results include details about subclasses of results and "near miss" elements in conjunction with positive results, the user can refine the query to find more appropriate results or widen the query to include more results if insufficient information is forthcoming. The third dimension allows the user a more expressive space, complete with navigation methods such as rotation and translation. In 3D, overlapping lines or labels can be avoided by rotating the layout to a better point of view.

DocuWorld (Einsfeld *et al.* 2006 [95]) is a prototype for a dynamic semantic information system. This tool allows computed structures as well as documents to be organised by users. Compared to the web Forager (Card *et al.* 1996[61]), a workspace to organise documents with different degrees of interest at different distances to the user, DocuWorld provides the user with more flexible possibilities to store documents at locations defined by the user and visually indicates cluster-document relations (different semantics of connecting clusters to each other). A detailed comparison of these techniques is presented in Table 3.7.

System	**Visual Representation**	**Interaction techniques**	**Interaction actions**			**Input-Output devices**	**3D/VR**	**year**
			Navigation	**Selection and Manipulation**	**System control**			
SUMO Buntain 2008 [53] (Figure 3.10)	Tree	Visual exploration	Manual view point manipulation	-	-	2D Mouse + 2D screen	3D	2008
DocuWorld Einsfeld *et al.* 2006 [95] (Figure 3.9(c))	Graph	human- centred	Thought wizard metaphor	Object selection + object positioning + virtual pointer	Gestural interaction + voice commands	Mouse + stereoscopic	VR	2006

Table 3.7: 3D VDM tool summary for classification KDD task

3.4.2.3 Association rules

On account of the enormous quantities of rules that can be produced by DM algorithms, association rule post-processing is a difficult stage in an association rule discovery process. In order to find relevant knowledge for decision-making, the user needs to rummage through the rules.

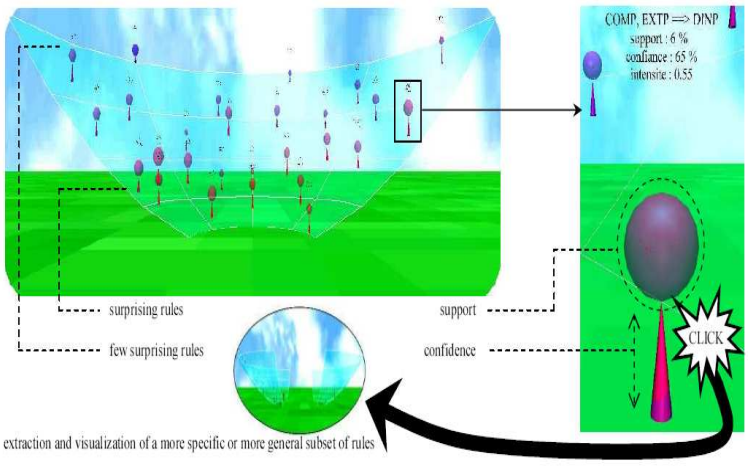


FIGURE 3.17: ArVis : a tool for association rules visualisation [31].

ArVis proposed by Blanchard *et al.* 2007 [31] is a human-centred approach. This approach consists of letting the user navigate freely inside the large set of rules by focusing on successive limited subsets via a visual representation of the rules (Figure 3.17). In other words, the user gradually drives a series of visual local explorations according to his/her interest for the rules. This approach is original compared to other rule visualisation methods (Couturier *et al.* 2007[79], Ertek and Demiriz 2006 [99], Zhao and Liu 2005[309]). Moreover, ARVis generates the rules dynamically during exploration by the user. Thus, the user's guidance during association rule post-processing is also exploited during association rule mining to reduce the search space and avoid generating huge amounts of rules.

Gotzelmann *et al.* 2007 [130] proposed a VDM system to analyse error sources of complex technical devices. The aims of the proposed approach is to extract association rules from a set of documents that describe malfunctions and errors for complex technical devices, followed by a projection of the results on a corresponding 3D model. Domain experts can evaluate the results gained by the DM algorithm by exploring a 3D model interactively in order to find spatial relationships between different components of the product. 3D enables the flexible spatial mapping of the results of statistical analysis. The visualisation of statistical data on their spatial reference object by modifying visual properties to encode data (Figure 3.12 (a)) can reveal a priori unknown facts, which where hidden in the database. By interactively

exploring the 3D model, unknown sources and correlations of failures can be discovered that rely on the spatial configuration of several components and the shape of complex geometric objects. A detailed comparison of these techniques is presented in Table 3.8.

System	**Visual Representation**	**Interaction techniques**	**Interaction actions**			**Input-Output devices**	**3D/VR**	**year**
			Navigation	**Selection and Manipulation**	**System control**			
ARVis Blanchard *et al.* 2007 [31] (Figure 3.17)	Geometric shape	Human- centred	Manual view point manipulation	Object selecting + virtual pointer	Graphical menus	Mouse + 2D screen	3D	2007
3D spatial data mining on document sets (Gotzelmann *et al.* 2007 [130])	Virtual world	Visual navigation and visual manipulation	-	Object selection	Graphical menus	-	3D	2006

TABLE 3.8: 3D VDM tools: summary for association rules in KDD tasks.

3.4.2.4 Combination of methods

The Haiku tool (Figure 3.9(b)) proposes combining different DM methods: clustering, classification and association rules (Beale *et al.* 2007 [20]). In this tool, the use of 3D graphs allows the visualisation of high-dimensional data in a comprehensible and compact representation. The interface provides a large set of 3D manipulation feature of the structure such as zooming in and out, moving through the representation (flying), rotating, jumping to specific location, viewing data details, and defining an area of interest. The only downside is that the control is done using a mouse. A detailed presentation is shown in Table 3.9.

System	**Visual Representation**	**Interaction techniques**	**Interaction actions**			**Input-Output devices**	**3D/VR**	**year**
			Navigation	**Selection and Manipulation**	**System control**			
Heiku Pryke- and and Beale 2005 [230] (Figure 3.9(b))	Graph	Human-centred	Manual view point manipulation + target based	Object selection	-	Mouse + 2D screen	3D	2005

TABLE 3.9: 3D VDM tool : combination of methods.

3.5 Conclusion

In this chapter, we have proposed a new classification of VDM tools composed of 3 dimensions: visual representations, interaction techniques, and data mining tasks. We have also proposed a survey of visual representations and 3D interaction techniques in data mining and virtual reality. Compared to 2D representation (presented in Chapter 1 Section 1.5.3) 3D representation allows the display of large data set but unlike 2D representation of association rules, 3D metaphors do not allow visualisation and manipulation of rule items. Therefore, the user can not directly find the rules that contain a given item or he/she can not require the presence of an item in the extracted rules.

Through this study, we can notice that most of the pre-processing tools use immersive configuration (CAVE) which is not the case of post-processing tools that still rely on interaction metaphors and devices developed more than a decade ago. VDM is inherently cooperative requiring many experts to coordinate their activities to make decisions. Thus, desktop configuration does not allows collective work like large-scale immersive configuration. We can also see that most tool are suitable only for data/knowledge exploration and don't offer manipulation techniques despite the fact that user participation in the process of data mining is paramount.

Now, it is questionable whether these classical interaction techniques are able to meet the demands of the ever increasing mass of information or whether we are losing because we still lack the possibilities to properly interact with the databases to extract relevant knowledge. Devising intuitive visual interactive representations for data mining and providing real time interaction and mapping techniques that are scalable to the huge size of many current databases are some of the research challenges that remain to be addressed.

4

Interactive Extraction and Exploration of Association Rules

Contents

4.1 Introduction

To handle the large amount of rules produced by the data mining algorithms, many solutions have been proposed in the literature, among them the post-processing of association rules. This consists of a second search operation with the aim of finding interesting association rules. Whereas the first operation (data mining) is done automatically by combinatorial algorithms, the search for interesting rules is generally left to the user.

In 1996, Brachman *et al.* 1996 [45] have pointed out that in order to efficiently assist the users in their search for interesting knowledge, the data mining process should be considered not only from the point of view of the discovery algorithms, but also from that of the user, as a human-centred decision-support system. For post-processing of association rules to be effective, we need to include the user in the KDD process and combine human flexibility, creativity and knowledge with the enormous storage capacity and the computational power of today's computers. Thus, interactive, efficient tools have to be developed.

In most association rule post-processing approaches, it is often through visualisation that rule post-processing is performed. Effective interactivity does not involve only interaction with the graphic representation of rules but with the data mining process itself. However, in most approaches, interactivity is not present throughout all the process of association rule generation. Figure 4.1 shows the different steps of the association rule generation process. The different figures represented in this diagram show the moments in which experts are involved in the association rule generation process. Even so, the association rule generation process must be able to rely on notions of task-oriented systems as defined in human computer interaction (Diaper 2003 [91], Greenberg 2003 [125]). In order to find relevant knowledge in visual representations, the decision-maker needs to freely rummage through large amount of data. It is therefore essential to integrate him/her in the data mining process through the use of efficient interactive techniques.

In this chapter, firstly we propose a new methodology for the interactive visualisation of association rules designed for the convenience of the user faced with a large set of rules, this being based on the principle of the cognitive processing of information. Section 2 is dedicated to a our association rules metaphor based on information visualisation principles for effective visual representations. In Section 3, we present the IUCEARVis functions that were used. For this, we refer to the visualisation process model proposed by Card *et al.* 1999 [60] for new visualisation tools that highlights three components: Data transformation, Rendering Mappings and View transformation.

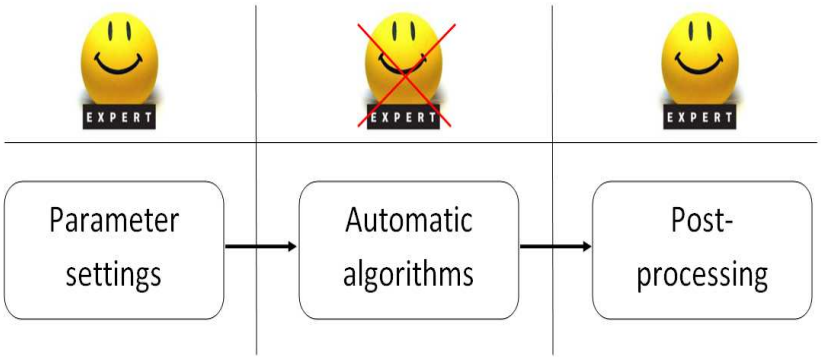


FIGURE 4.1: Expert role in the association rule generation process

4.2 Constraints of the Interactive Post-processing of Association Rules

During the post-processing step, the user is faced with large set of rules. In order to find interesting knowledge for decision making, he/she needs to (*i*) interpret the rules in its business semantics and (*ii*) evaluate their quality. The decision-making indicators are (*i*) the syntax of rules (the items involved in each rule) and (*ii*) the association rules interestingness measures.

In this context, the user's task can be extremely difficult for three main reasons:

- 3D association rule representations does not allow the visualisation of the rule syntax.
- the huge number of rules at the output of the association rules extraction algorithms makes their exploration a very painful task;

- association rule extraction is a non-supervised process. Association rules are typically used when the user does not know what he/she is looking for. Therefore, the user can not formulate constraints that would directly isolate interesting rules. Usually, the user is looking for knowledge that he/she does not expect.

4.2.1 Importance of the Individual Attributes of Rules

All techniques proposed for rule visualisation in VR (Chapter 3 Section 3.4.2.3) have been developed to represent an association rule as a hole without paying attention to the relations among the attributes that make up the antecedent and the consequent and the contribution of each one to the rule. Attribute components of an association rule and their contribution to the rule may be more informative than the rule itself (Freitas 1998 [109]). Two rules with the same value of rule interestingness measure can have very different degrees of interestingness for the user, depending on which attributes occur in the antecedent and in the consequent.

In the same way, the information found in form of relations between the attributes (correlation) provides the analyst with a better and clearer image than analysing a rule (Imielinski 1998 [152]). Exploring an association rule attribute enables deeper insight into the data. Analysts can be interested by these relationships, rather than a static rule. Many researchers have demonstrated that attribute interaction is a key concept of association rule mining (Freitas 2001 [108], Chanda *et al.* 2010 [67], Yao *et al.* 1999 [301]).

4.2.1.1 Attribute importance

An attribute can be important for the user if regularities are observed in a smaller dataset, although being unobservable in the entire data. A rule can be considered as a disjunction of rules. The size of a disjunct (rule) is the number of items composed in the rule's antecedent and the rule's consequent. For example, r : ($milk$, $bread$, $eggs \rightarrow apples$, $pears$) is a rule. A disjunction of a rule is $r1$: ($milk \rightarrow apples$, $pears$), $r2$: ($bread \rightarrow apples$, $pears$), r3 : ($eggs \rightarrow apples$, $pears$), $r4$: ($apples \rightarrow milk$, $bread$, $eggs$), and $r5$: ($pears \rightarrow milk$, $bread$, $eggs$). At first sight, it seems that these small rules have no importance, since they can be considered as redundant rules. Based on this view, most extraction algorithms do not keep these rules in the results. However, small rules have the potential to show unexpected relationships in the data (Freitas 1998 [109]). Provost and Aronis 1996 [229] proved that small rules were considered interesting in their field application. Accordingly, it would beneficial that the user see automatically these small rules.

In order to evaluate the contribution of each item to a rule, Freitas 1998 [109] proposed the *Information Gain* measure which can be positive or negative. The *Information Gain* measure (Freitas 1998 [109]) was proposed for rules with only one

item in the consequence. In our representation, if an *item* $\in$ *Antecedent* has respectively a high positive (negative) *Information Gain* that mean that: r : (*item* $\rightarrow$ *Consequence*) is a good (bad) rule whatever the number of items in the consequent. From a rule interest perspective, a high *Information Gain* may point out new interesting implications unknown by the user. At the same time, a rule including attributes with low or negative information gain indicate irrelevant rules. Therefore, the user does not waste time looking at these rules since he/she already knows that they are not interesting.

4.2.1.2 Attribute interaction

Two attributes are correlated if they are not independent. Two attributes are independent if changing the value of one does not affect the value of the other. The lift measure calculates the correlation between each attribute pair from the antecedent or the consequent. The correlation between two attributes represents the amount of information shared between the two attributes. The lift measure determines whether two attributes are correlated (lift >1) or not (lift <1) (see Chapter 1 Section 1.3.2).

The Freitas 2001 [108] study showed that the concept of attributes interaction can be beneficial to the association rule extraction process and proposed to introduce attribute interaction in the design of association rule mining systems. Attribute interaction enable the detection of surprising knowledge which can't be discovered by analysing the whole rule. The relationships expressed in a rule totality is quite different from the relationships expressed in separate rule parts (antecedent and consequent).

On the other hand, to discover useful association rules, the user needs to get insight into the data and learn about the data model by exploring attribute relations (Chanda *et al.* 2010 [67]). In many case (a biological or genetic context for example) antecedent items has weak associations with consequent. However, they interact together in a complicated way to control the consequent (Chanda *et al.* 2010 [67]).

4.2.2 Hypothesis About The Cognitive Processing of Information

Based on the bounded rationality assumption (Simon 1979 [251]), a decision-making process can be regarded as seeking a dominant structure. More precisely, the user faces a set of options about using rules as operators in this search for dominant structure. This dominant structure is a cognitive structure in which an alternative choice can be seen as dominant over the others (Montgomery 1983 [200]). This decision making model can be used for association rule post-processing. According to Montgomery 1983 [200], the user isolates a limited number of potentially interesting rules and performs comparisons among them. This is done several times during the decision making process with the aim of finding rules that are more interesting then the other ones. Considerably early in a decision making process, the decision maker tries to find promising rules, which can be replaced by more interesting ones during the

process. These changes can take place several times during the process, particularly when the user has trouble to find dominant structures.

Bhandari 1994 [25] proposed a *machine-assisted knowledge discovery* approach. A computer program guides a decision maker to discover knowledge from data. This approach was based on experimental data related to user behaviour during a decision making process study. The *machine-assisted knowledge discovery* approach is based on a new KDD methodology called *attribute focusing*. In this methodology, an automatic filter detects, using statistical measures, a small number of potentially interesting items. *Attribute focusing* aims to point out small and thus more comprehensible subsets of items. Later, Hahsler *et al.* 2007 [134] proposed a two-step approach called *interesting itemsets* (e.g., by a mixture of mining and expert knowledge). In the first step, the user defines a set of interesting itemsets and then, in a second step the system generates rules based only on these itemsets. The interest to target a small number of items in the cognitive processing of information has also been strongly confirmed by strategy-making work (see Barthelemy 1992 [16] for more information). In fact, because of the human's limited cognitive capabilities, the user can examine at a given moment of time only a small amount of information. Therefore, the decision maker should be able to save the interesting association rules, discovered from different subsets of rules, and compare them.

Today, KDD is viewed as a session-driven activity. Therefore, the mining results are typically displayed on the screen, viewed by the user and subsequently, completely discarded. Thus the typical lifetime of the discovered knowledge is the duration of the mining session. There is little or no support for the user to rely on the results of the previous mining sessions. In such a system, the discovered knowledge from possibly many mining sessions can be stored in memory. Furthermore, one can build upon the knowledge accumulated over time.

From these various works on the cognitive processing of information, we established five principles underlying our methodology for the post-processing of association rules.

- *P1*: lets the user choose a limited number of potentially interesting items;
- *P2*: provides the user with association rules that may be interesting;
- *P3*: allows the user to make comparisons among association rules;
- *P4*: allows the user to modify the subset of potentially interesting items at any time during post-processing;
- *P5*: allows the user to note the interesting association rules and save them with notes.

4.3 IUCEAR: Methodology for Interactive User-Centred Exploration of Association Rules

We assign R, the complete set of association rules produced by an exhaustive algorithm for association rule extracting. R contains most of the time a large number of association rules. Our methodology for association rule post-processing is designed for the convenience of the user faced with a large size of association rule set R. It is necessary to let the user navigate in R by exploring limited subsets of association rules. The association rules and their interesting measures are presented in graphic form. This means that the user controls, through successive trial and error, a series of local visual explorations based on his/her interest in the rules (Figure 4.2).

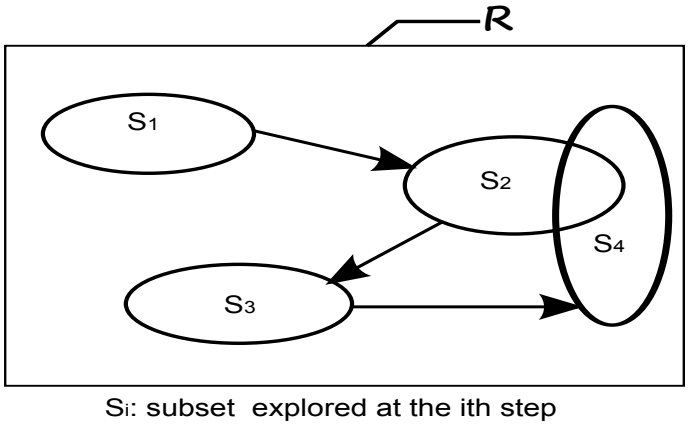


FIGURE 4.2: Exploration of limited subsets of association rules in R

Thus, the set R is explored subset after subset. The user never faces all association rules at the same time. At each navigation step, the system proposes to the user a set of association rules that can interested him/her. Then, the user may find an interesting association rule and decide to keep it in the memory or may take the decision to choose the next association rules subset to visit. It is through this process, that the user subjectivity is expressed in the association rules post-processing. *Filtering functions* can be used to prune the association rule set proposed by the system.

Our methodology includes the guiding principles of the cognitive processing of information presented in Section 4.2.2 as follows:

1. functions to select a subset of potentially interesting items (*P1*);
2. relations to target subset of association rules and to navigate among them (*P2* and *P4*);
3. the user views subset of rules (*P3*). The visualisation simplifies comparisons among pieces of information (Ceglar *et al.* 2003 [63]);

4. the user saves the association rules that seem interesting (*P3* and *P5*).

The functions, relations and visualisation must take into account the two indicators involved in the user task: the syntax of the rules and the interestingness measures of the values.

4.3.1 Items Selection

Item Selection is the first step of the proposed methodology. We take advantage of the fact that the user focuses on subsets to display only the subset being explored. Thus, the functions ***add to the antecedent*** and ***add to the consequent*** perform the data transformation from a set of items I to a rule A.

Through this step the user selects a set of interesting items (P1). The function is defined as follows: in the whole item set I, two relations ***add to the antecedent*** and ***add to the consequent*** associate each selected item to the antecedent or to the consequent (Figure 4.3) of a *reference rule* (used to target a subset of association rules). To add an item, the user must make two choices: which relation to apply and on which item.

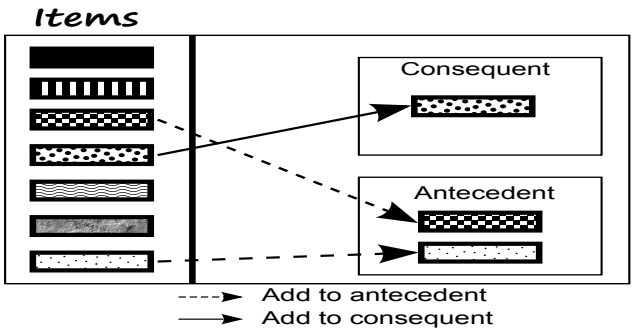


FIGURE 4.3: Each relation adds a selected item to the antecedent or to the consequent.

4.3.2 Local mining: anticipation functions

Anticipation functions perform the transformation of a rule A to a set of rules S by targeting subsets of rules. Interactive operators must be integrated into the visualisation process to allow the user to activate these functions.

These functions define how subsets of rules are constituted from items selected in the previous step. As a vector of user navigation, those functions are fundamental elements of the IUCARE methodology. The anticipation functions are defined as follows: in all sets of rules R, the anticipation functions associate each rule to a limited

subset of rules that could be interesting to the user (Figure 4.4).

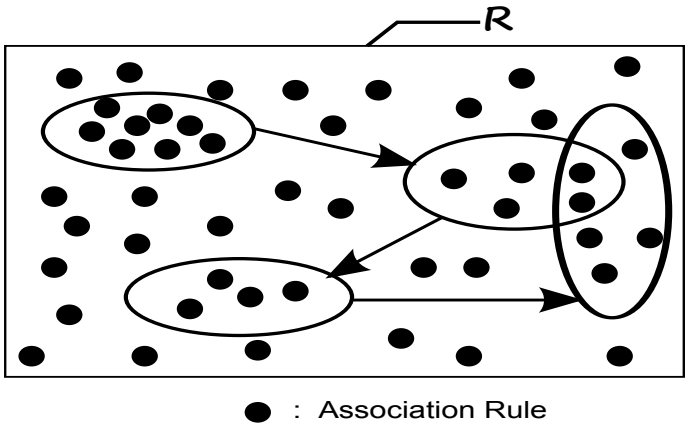


FIGURE 4.4: Anticipation functions associate each association rule chosen or constructed by the user to a subset of rules.

From a rule the system proposes x rules. To navigate from one subset to another, the user must choose which rule to apply the anticipation functions (from the proposed x rules) otherwise he/she can always change the selected items (Figure 4.5).

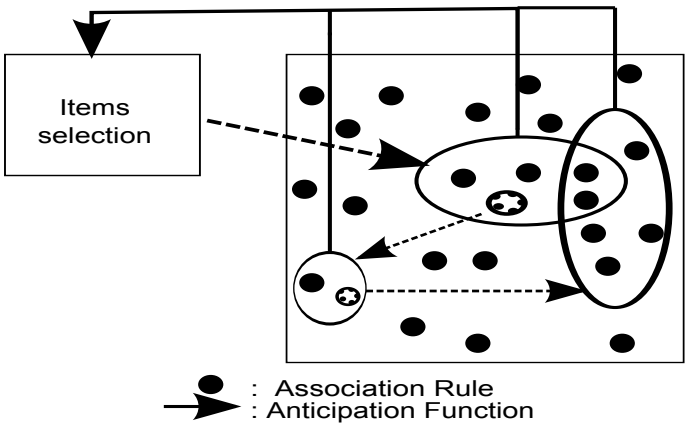


FIGURE 4.5: To navigate from one subset of rules to another, the user can choose one rule from the current subset of rules or change the selected items.

$\forall(r_1,r_2) \in R^2, (r_2) is-anticipation-rule-of(r_1) \Leftrightarrow$(the system determines that r_2 is interesting for the user based on r_1). Here, the 5 anticipation functions $(r_2)$is-anticipation-rule-of(r_1):

- r_2 is an anticipation rule of r_1 if, and only if, r_2 has a more general antecedent than r_1 – and has an item more in the antecedent.
- r_2 is an anticipation rule of r_1 if, and only if, r_2 has a more general consequent than r_1 – and has an item more in the consequent.

- r_2 is an anticipation rule of r_1 if, and only if, r_2 has an item less than r_1 in the antecedent.
- r_2 is an anticipation rule of r_1 if, and only if, r_2 has an item less than r_1 in the consequent. If there is only one item in the antecedent or in the consequent, this item will be replaced by another item from the database.
- r_2 is an anticipation rule of r_1 if, and only if, r_2 has the same antecedent of r_1 and a different consequent .
- r_2 is an anticipation rule of r_1 if, and only if, r_2 has the same consequent of r_1 and a different antecedent.

All functions are based on rules syntax (Figure 4.6).

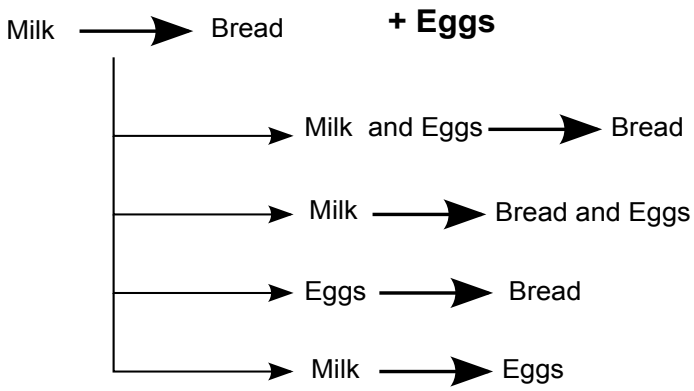


FIGURE 4.6: Illustration of the anticipation functions.

From all the items of the database, the number of rules constructed by the anticipation functions can be very high. The larger the number of items in the database, the greater the number of proposed rules. But as we have already mentioned, the user can not analyse a large number of association rules. Therefore, association rule ranking functions are used to propose only a limited number of rules to the user at a time. These functions use the rule's interestingness measures to rank the set of rules before issuing them. The user visualises only a small set of high-ranking rules at a time. The originality of the IUCARE methodology in comparison with other methods of rule exploration (presented in Section 1.5.3.2) mainly lies in the concept of anticipation and ranking functions. The user can elaborate hypotheses during the navigation process, unlike an association rules explorer or a query language, since the user must lay down his hypothesis before starting the process of extracting association rules. Using a rules explorer or a query language, the extraction constraints must be explicitly set, whereas with the IUCARE methodology, constraints are expressed implicitly as they are integrated into the anticipation functions. We believe that these functions facilitates the user task by making constraint specification much easier and offer knowledge based on the his/her interest because it does not require the user to know what he/she looking for.

Example 4.3.1 Consider the following scene in an association rule post-processing process: The user thinks that the association rule $milk$, $bread$, $eggs \rightarrow pears$ is interesting but he/she does not know that $milk$, $bread$, $appels \rightarrow eggs$ and $milk$, $bread \rightarrow jus$ are even more interesting. With the IUCARE methodology, the user has access to valuable information that he/she does not know and that are based on his/her intentions and preferences. However, if the user uses a query language to perform post-processing rules, then it will not account for the existence of these interesting rules because he did not know beforehand. If he/she uses a rules explorer, then he/she must find each rule manually via the graphical interface. If the user chooses Arvis (Blanchard *et al.* 2007 [31]) to explore association rules post-processing, he/she needs to use two different neighbourhood relationships to achieve the two rules. In the 3 cases, the user's task can be tedious.

4.3.3 Association Rule Visualisation, Validation, and Evaluation

The IUCEAR methodology aims to help the user find interesting rules. For that, the user should be able to evaluate and to compare the association rules proposed by the anticipation functions.

Visualisation can be very beneficial to association rule mining (Simoff *et al.* 2008 [250]). In fact, visualisation techniques are an effective means to provide users with meaningful visual representations instead of poorly intelligible textual lists. Visualising rules makes it easier to find rules that have high interestingness measures and to compare them (cognitive principle *P2*). A significant flaw in most visual representations presented in Chapter 1 Section 1.5.3 such as grid-like structures and bar charts it is that it does not highlight the rule's interestingness measures although this information is crucial to rule validation. For example, matrix visualisation, graph visualisation, and parallel coordinate visualisation use colour to represent some rules interestingness measures, even though this graphic encoding for quantitative variables is known to be bad in information visualisation. The use of colour to represent quantitative ordinal variables is very often tolerated when it is used to represent a few modalities (Wilkinson 2005 [291], Spence 2001 [256]). This can especially be applied to quantitative variables such as rule interestingness measures if we accept to discretise them. Strictly speaking, such a solution should be rejected because the colours do not induce any universal order, and their application is unique each time (Bertin 1969 [24], Mulrow 2002 [202]).

4.3.4 Browsing History

Rule exploration may be facilitated by providing the user with a rule navigation card. During the association rule exploration process, the user can show his/her interest to some association rules. These association rules are stored as a navigation map. This navigation card can take the form of a graph indicating the score for each association rule and the order of their registration (Figure 4.7). This representation would serve as historical exploration but also to give the user an overview of all the rules he/she

has visited and that have attracted his/her attention.

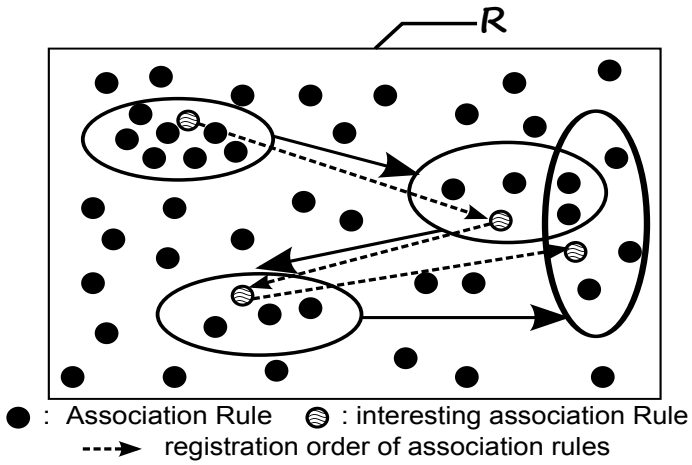


FIGURE 4.7: Illustration of rules navigation card.

4.3.5 Interactive process

The interactive process of the IUCEAR methodology is presented in Figure 4.8. The process aims at guiding the user through the extraction and post-processing of rules. First, the user should build a *reference rule*. Then, taking into account the system propositions, he/she is able to revise his/her expectations and actions. Functions, relations and visualisation interact as presented in the Figure 4.8.

4.4 New Association Rules Metaphor

As we have seen in Section 4.2.1, attribute components of an association rule a may be more informative than the rule itself (Freitas 1998 [109]). In this context, we propose a new visualisation metaphor for association rules. This new metaphor represents attributes which make up the antecedent and the consequent, the contribution of each one to the rule, and the correlations between each pair of the antecedent and each pair of consequent. This metaphor is developed in 3D to overcome some limitations of 2D representations.

4.4.0.1 Rendering Mapping of Association rule metaphor

The association rule representation proposed is based on a molecular metaphor. The association rule representation is composed of several spheres. As advocated by Bertin 1969 [24], we chose graphic encoding based on positions and sizes to enhance the most important interestingness measures, these being: *Information Gain* and correlation between attributes. To have the greatest degree of freedom we chose to use a 3D

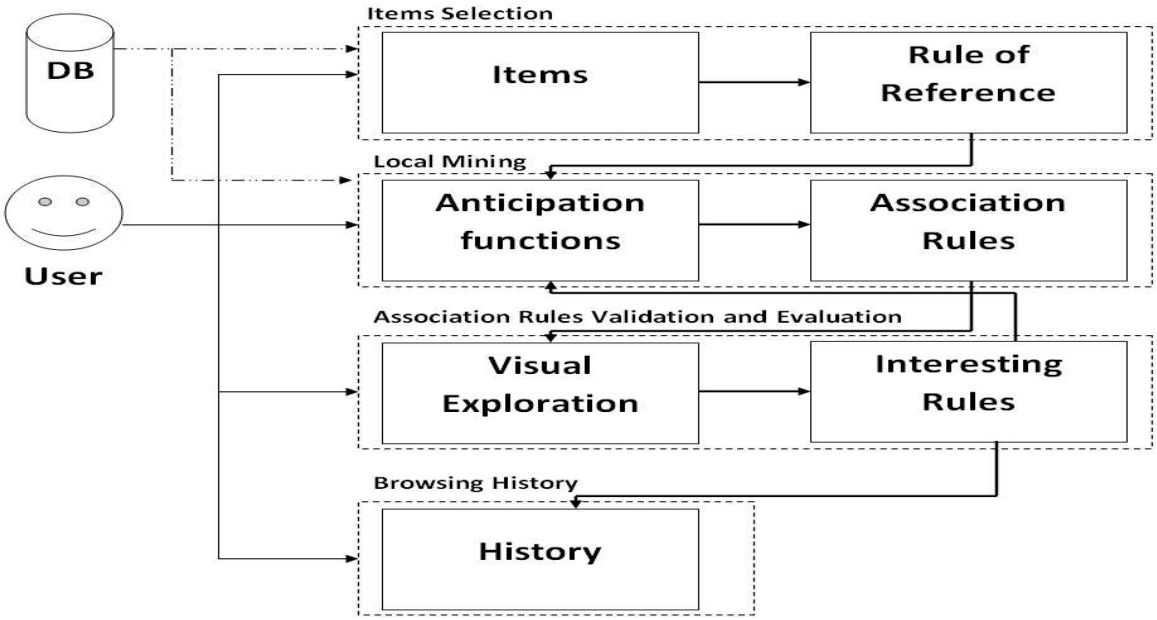


FIGURE 4.8: Interactive process description for the IUCEAR methodology.

representation.

Our metaphor (Figure 4.9) shows two types of interestingness measures.

The first one matches rule attribute description which consists of categorical variables from Bertin's semiology point of view. Each attribute has an associated continuous variable corresponding to its *information gain*. The user must know if an item belongs to the antecedent or the consequent. Therefore, we should separate the items of the antecedent from the items of the consequent in the representation space. Each sphere represents an item and its size and colour represent its contribution to the rule. The sphere size represents the contribution of the item and the colour shows if the contribution is positive (blue) or negative (grey). The chosen graphical encoding highlights items with high, positives contribution (large blue sphere) and those with high, negative contributions (large grey sphere). Both sets of information are interesting to the user.

The lift is a positive measure used to indicate to what degree two items of the antecedent or two items of the consequence are correlated. A distance between each two items of the same side (antecedent or consequence) is an effective representation of this measurement. The more the items are correlated, the closer the spheres. In our representation, the antecedent and the consequence are two separate graphs in which the nodes are items. To generate item coordinates in 3D space, we use a modified version of the spring-embedded like algorithm (Hendley *et al.* 1999 [141]).

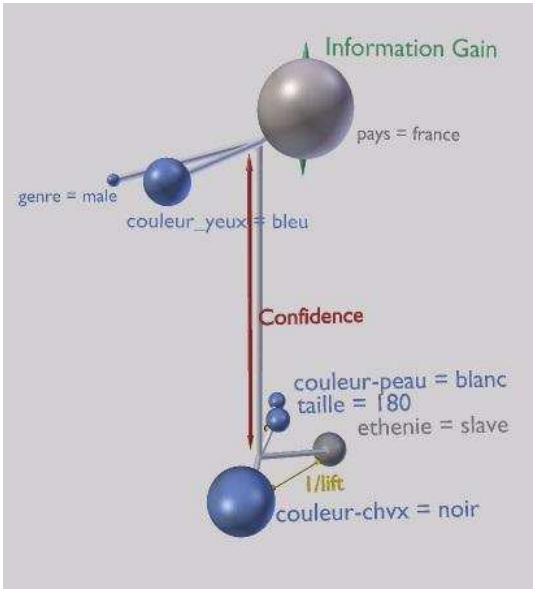


FIGURE 4.9: The visual association rule metaphor.

This algorithm (see Chapter 3 Section 3.3.1.1) enables the spheres self-organisation in the visualisation area by the use of a force system in order to determine the sphere positions. Using a force system allows correlated items to be near each other, and independent items to be far away. The algorithm is described in the following section. An association rule representation consists of spheres and links whose properties are given by the rule parameters.

The second type of interestingness measure corresponds to measurements associated with the rule (support, confidence, etc,.). This meta information that describes the properties of the rules are quantitative variables according to Bertin's semiology 1969 [24]. Theoretically, it is possible to represent a large number of metrics using visual variables appropriate to the area of interest of each user. For example, we can represent the confidence or the support by a distance between the antecedent and the consequence. The visual metaphor stresses the rules with a high level of confidence or support (Figure 4.10). Furthermore, complementary text labels appear above each object to give the name of the corresponding item.

4.4.0.2 Spring-embedded like algorithm

The graph representing the antecedent items or the consequent items is projected into a 3D space. Each graph node represents an item. Nodes are represented by spheres. The node coordinates are determined by a system of attractive forces (Table 4.4.0.2).

The nodes are all connected with edges. The edge is considered as a spring between two nodes with length and elasticity attributes. The spring length represents

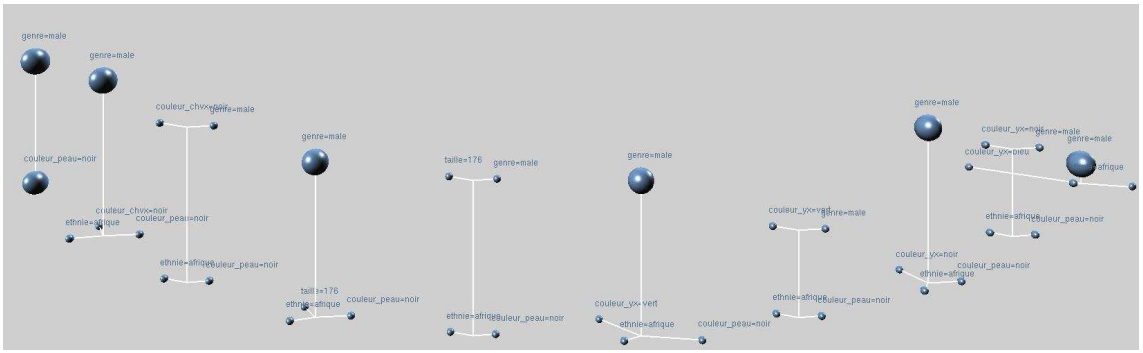


FIGURE 4.10: Illustration of an association rules set. The distance between the antecedent and the consequence stresses the rules with a high interestingness measure (support of confidence)

Input: Set of item l, Set of Edge E, Force = (x=0,y=0,z=0)
Output: Set of coordinates C

1. **forall** item $l_k \in l$ **do**
5. **forall** Edge $E_m \in E$ connected to l_k**do**
6. Force = Force + HookeAttraction(l_k, E_m)
7. **endfor**
8. Velocity = (Velocity + Force) * damping
9. $C = C$ * Velocity
10. **endfor**

TABLE 4.1: The placement algorithm.

the correlation between the two items (1/lift). More the items are correlated, the more the edge are short and the closer the nodes. To reduce the number of constraints, we consider only relevant links whose lift is greater than 1 (see Chapter 1 Section 1.3.2).

The *Hooke attraction (F_H)* describes the elasticity of the spring. It is applied between each pair of nodes and tends to maintain them at the defined distance (1/lift).

Definition 4.4.1

The Hook attraction F_H, applied to a node of coordinates C_1 linked to another node of coordinates C_2 where k is the virtual stiffness and R the length of the edge.

$$F_H = -k\frac{(|L-R)L}{|L|} \; with \; L = p_2 - p_1$$

The system produces a physical approximation of the node movement which can be easily interpreted by a human where the nodes are distributed in space according to the edge length (correlation among the items).

4.4.1 Validation of Association rule metaphors

In this section, we present the results of the evaluation of four different representations of association rules with different interaction techniques and visualisation configuration. The four representations considered in this study are based on the metaphor presented in the previous section. This experiment aims to evaluate the four different representations in different configurations. In this study, only one association rule is visualised by users and the different association rules are only one item in the consequent.

- Metaphor 1 (Figure 4.11(a)): the items are placed in a circle with an ascending *Information Gain*;
- Metaphor 2 (Figure 4.11(b)): the items are placed using the spring-embedded like algorithm;
- Metaphor 3 (Figure 4.11(c)): the items are placed in a circle in pairs with an ascending *Information Gain*. The item with the highest *Information Gain* is placed first with the most correlated item next to it. The distance between the two items is proportional to the lift value.
- Metaphor 4 (Figure 4.11(d)): the items are placed in a circle in pairs with an ascending *Information Gain* without the distance between the antecedent and the consequent.

4.4.1.1 Objective

The objective of this study is the determination of *i* the best metaphor to represent association rules, *ii* the more adequate visualisation interface and *iii* the best interaction technique, to offer the best performance and most suitable tool for the user.

4.4.1.2 Task

The task was to answer five different questions about each of the four displayed association rule representations.
The five questions are:

- Question 1: What is the item of the consequent?
- Question 2: What is the most influential item of the association rule?

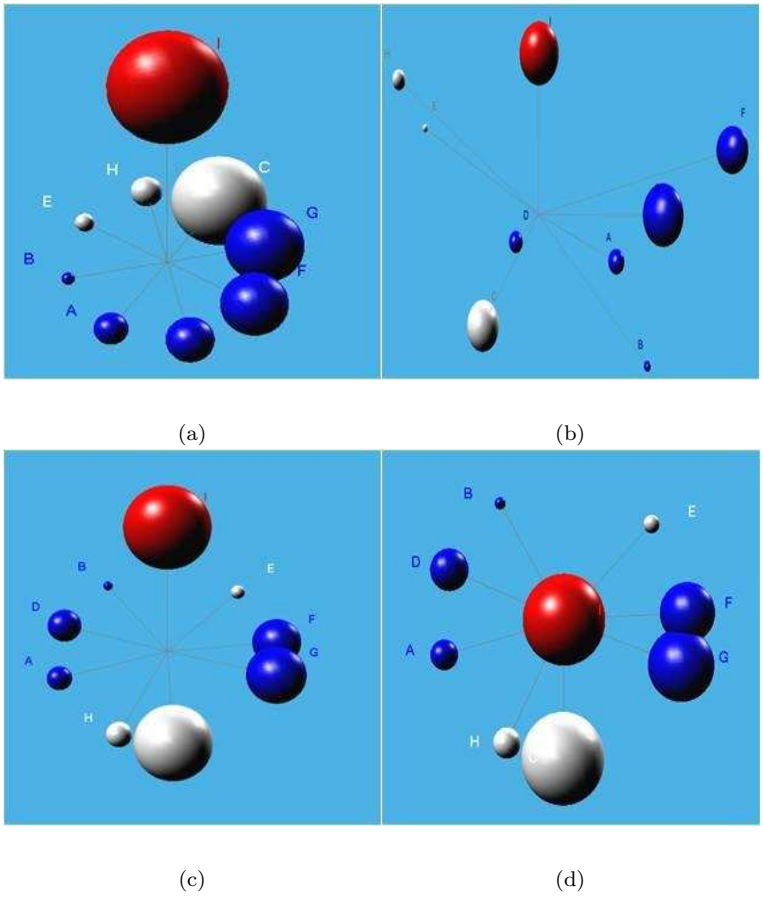


FIGURE 4.11: The 4 metaphors of association rule : (a) Metaphor 1 (b) Metaphor 2 (c) Metaphor 3 (d) Metaphor 4

- Question 3: What are the most correlated items of the association rule?
- Question 4: What are the lowest correlated items of the association rule?
- Question 5: What is the total number of antecedent items ?

The displayed association rules have four levels of complexity. Complexity is proportional to the number of items of association rules. There are four different association rules extracted from the database of Polytech'Nantes student grades between 2002 and 2005.

The different rules are:

- Association rule 1 (9 items): Social and Professional Issues(3rd)=B, Information systems(3rd)=C, Software Engineering(3rd)=B, Applied Mathematics(3rd)=B,

Foreign Languages(4th)=D, Applied Mathematics(4th)=B, Networking and Communications(4th)=B, Social and Professional Issues(4th)=A → Social and Professional Issues(5th)=B

- Association rule 2 (7 items): Software Engineering(2007)=E, Foreign Languages(2007)=C, Networking and Communications(2007)=C, Information Systems ID(2007)=D, Social and Professional Issues(2007)=D, Foreign Languages(2008)=B, Information Systems ID(2008)=D → Humanities and Professional Issues(2009)=C
- Association rule 3 (5 items): Foreign Languages=D, Social and Professional Issues=C, Application of Mathematics and Statistics to Decision-Aide=D, Knowledge-based Systems=D, Training Periods and Term Projects=C → Information Retrieval=C
- Association rule 4 (4 items): Social and Professional Issues=C, Information Systems=D, Systems and languages=D → Software Engineering=C

To simplify things, each item will be represented by a letter. Subjects should respond to a total of 20 questions (five questions for each one of the four association rules). Interaction with the representation is done using rotations on the 3-axis and a zoom (in/out).

4.4.1.3 Protocol

A total of 36 volunteer subjects, all right handed, participated in this study. These subjects had normal or corrected vision. Two protocols were used:

- Protocol 1: 24 of the 36 subjects performed the task only once, in one of the two conditions C_1 or C_2 (12 C_1 and 12 C_2);
- Protocol 2: 12 of the 36 subjects performed the task twice without a rest period between the two tests, in the conditions C_3 and C_4. 50% of the subjects began in condition C_3, then did the test in condition C_4 and 50 % of the subjects began in condition C_4, then do the test in condition C_3.

The experimental conditions are the following:

- C_1: wiimote^{TM} and stereoscopic display;
- C_2: wiimote^{TM} and monoscopic display;
- C_3: mouse and stereoscopic display;
- C_4: mouse and monoscopic display.

To get acquainted with the system, the subjects had to interact with a test association rule in the different conditions, before starting the test. The order of display of the different association rules was randomly selected for each subject to avoid bias

due to training transfer. The response time for each question was recorded. The subjects were equipped with stereoscopic active glasses in conditions C_1 and C_3 and were placed in front of the screen at a distance of 1.5 meters (Figure 4.12). To collect subjective data, a questionnaire was given to each subject at the end of the experiment.

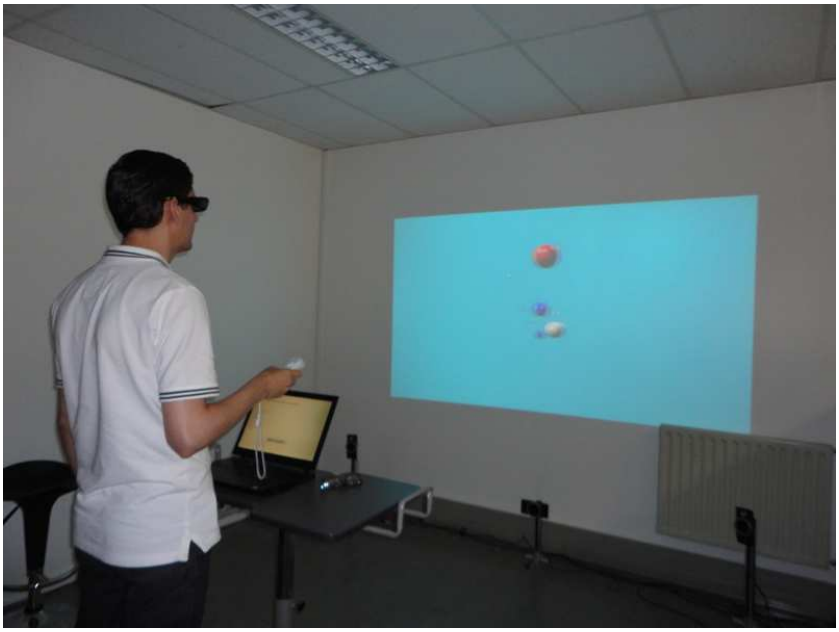


FIGURE 4.12: The test conditions.

4.4.2 Results

Firstly, we examined the task completion time and the error rate. Then, we examine the subjective aspects of performance (information collected via the questionnaire). We also reported the information collected (observation) during the experiment (difficulties encountered, strategies, subjects behaviour, etc.).

4.4.2.1 Response Time

Recorded data (response time) was processed through the analysis of variance (ANOVA).

Concerning the identification of the consequent item (question 1), the results shown in Figure 4.13 indicate that metaphor 4 has a better response time than metaphor 1 and metaphor 3 ($F(3,11) = 2.94, p = 0.04$). ANOVA found that metaphor 4 (M = 9.2s, SD = 3.1) allowed a quicker identification of the consequent than metaphor 1 (M = 14.3 s, SD = 6.2) and metaphor 2 (M = 14.9 s; SD = 8.0). No difference was found between metaphors 3 and 4. Therefore, we can deduce that metaphor 4 is best since the standard deviation is smaller. This result shows that the

user's performance is predictable.

For this task, monoscopic viewing is sufficient (Figure 4.14) and stereoscopic display does not help the user much . The interaction mode (computer mouse or Wiimote^{TM}) does not have a significant effect on performance ($F(3, 11) = 0.45, p > 0.05$) except in the case of using the Wiimote^{TM} with monoscopic viewing. In this case, ANOVA revealed better results than the stereoscopic display ($F(1, 22) = 5.53, p = 0.02$). ANOVA also revealed no difference between monoscopic and stereoscopic viewing when using the mouse only.

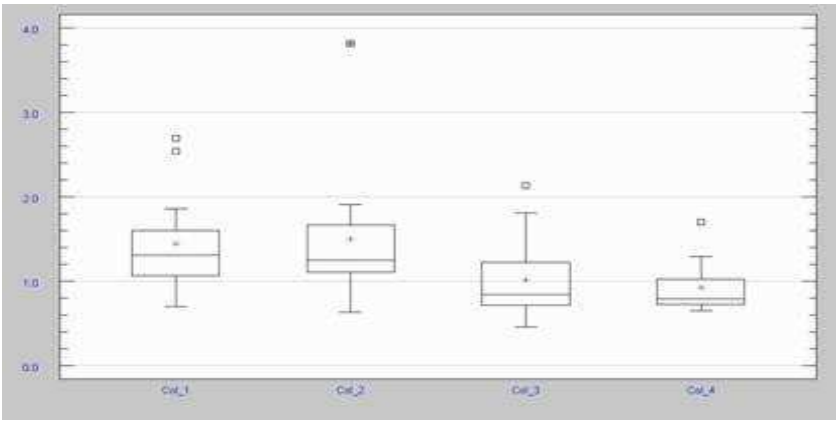


FIGURE 4.13: Response time to question 1 for different metaphors.

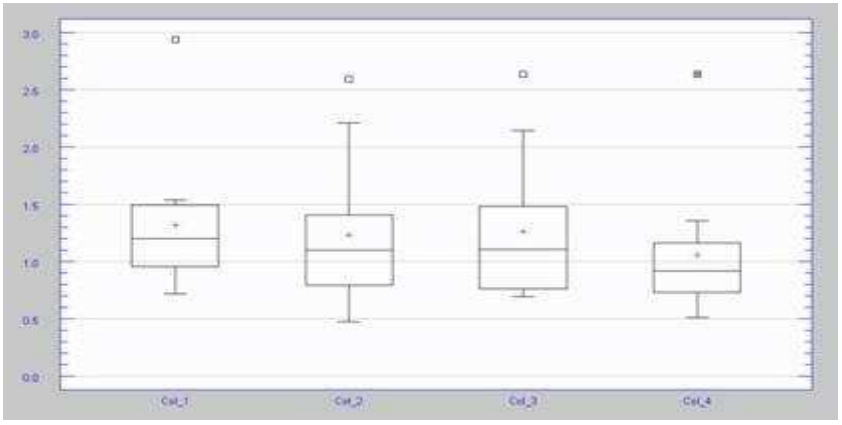


FIGURE 4.14: Response time to question 1 for different conditions.

To determine the dominant item of the association rule (Question 2), the results shown in Figure 4.15, indicate that metaphor 3 (M = 16.9 s, SD = 6.3) and metaphor 4 (M = 13.8s, SD = 2.8) are better compared to metaphor 2 (M = 22.7s, SD = 7.9) in terms of response time ($F(3, 11) = 4, 13, p = 0.01$).

The identification of the largest sphere does not require a specific method for viewing in the case of using the mouse (Figure 4.16) with monoscopic display (M = 15.08s, SD = 6.08) and stereoscopic display (M = 15.0s, SD = 5.67). The use of the Wiimote^{TM} (M = 21.69s, SD = 8.6) generates a response time higher than the mouse (M = 15.08s, SD = 6.08) in the case of monoscopic display ($F(3.11) = 3.06, p = 0.04$). To this question the mouse is better than the wiimote^{TM} only in the case of monoscopic viewing. In the case of stereoscopic viewing the mode of interaction does not influence the response time. ANOVA found no differences between monoscopic display and stereoscopic display with the mouse.

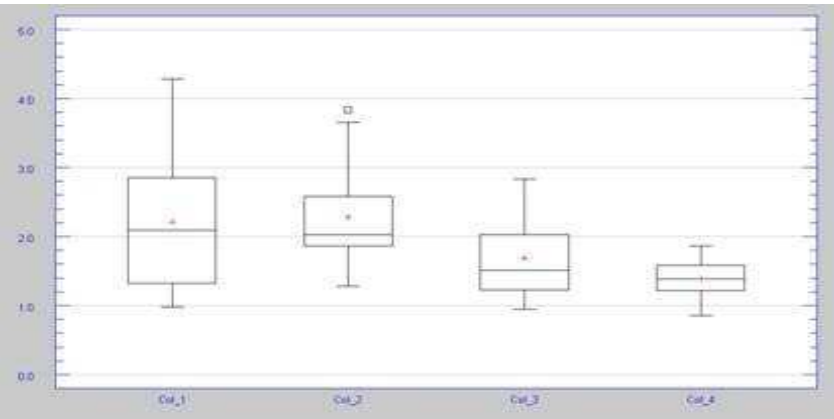


FIGURE 4.15: Response time to question 2 for different metaphors.

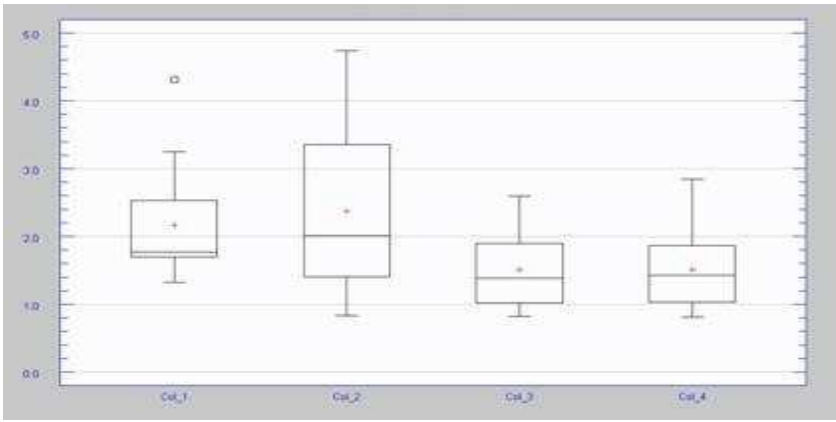


FIGURE 4.16: Response time to question 2 for different conditions.

The results of identification of the most correlated items ($F(3, 11) = 10.78, p = 0.00001$) shown in Figure 4.17 indicates that metaphor 3 (M = 19.55s, SD = 4.81) and metaphor 4 (M = 15.66 s, SD = 3.94) requires a response time lower than needed by metaphor 1 (M = 31.89s, SD = 14.96) and metaphor 2 (M = 31.81s; SD = 7.04). We can also assume a slightly better performance of metaphor 4 (M = 15.66 s, SD =

3.94) compared to the metaphor 3 (M = 19.55s, SD = 4.81) at the average response time and level of standard deviation.

ANOVA found ($F(3, 11) = 2.94, p = 0.04$) (Figure 4.17) that for this task using the configuration wiimote^{TM} / stereoscopic display (M = 31.46s, SD = 12.16) requires a response time greater than the configuration of mouse/stereoscopic display (M = 22.74 s, SD = 10.23). However, the display mode in itself is not involved, because we see no significant difference between the use of stereoscopy or monoscopy with the mouse. For this task, unlike question 2, using the wiimote^{TM} required a longer response time than the case of stereoscopic viewing (for question 2 it was in the case of monoscopic display). For monoscopic display the mode of interaction is not involved. As for question 1, ANOVA (F (1,22) = 4.50, p = 0.04) found that the configuration wiimote/stereoscopic display had a response time greater than the configuration wiimote/monoscopic display. ANOVA found no differences between monoscopic display and stereoscopic display with the mouse.

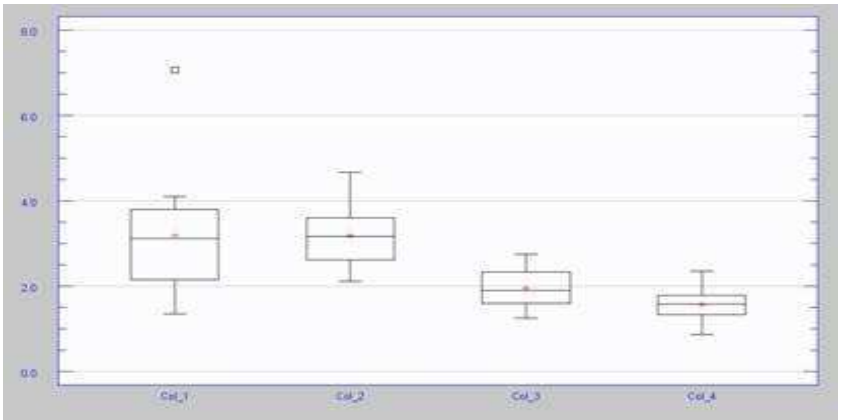


FIGURE 4.17: Response time to question 3 for different metaphors.

For the identification of the least correlated item of an association rule, the results ($F(3, 11) = 3.80, p = 0.02$) shown in Figure 4.19 indicates that metaphor 3 (M = 19.58s, SD = 7.69) and metaphor 4 (M = 17.10s, SD = 5.01) are better compared to metaphor 1 (M = 27.48s, SD = 11.38).

For the test conditions, as for other tasks, the Wiimote^{TM} /stereoscopic display configuration requires a greater response time compared to other conditions. ANOVA (Figure 4.20) ($F(3, 11) = 3.88, p = 0.02$) revealing that using the Wiimote^{TM} as an interaction tool in monoscopic display (M = 21.s; SD = 9.55) is better than stereoscopic (M = 28.40, SD = 11.71). According to previous tasks, ANOVA revealed ($F(1, 22) = 6.15, p = 0.02$) that the Wiimote^{TM}/stereoscopic display configuration needs a response time greater than the Wiimote^{TM} /monoscopic display configuration. ANOVA did not reveal a significant difference between monoscopic and stereoscopic viewing using the mouse. For the last task (identifying the total number of

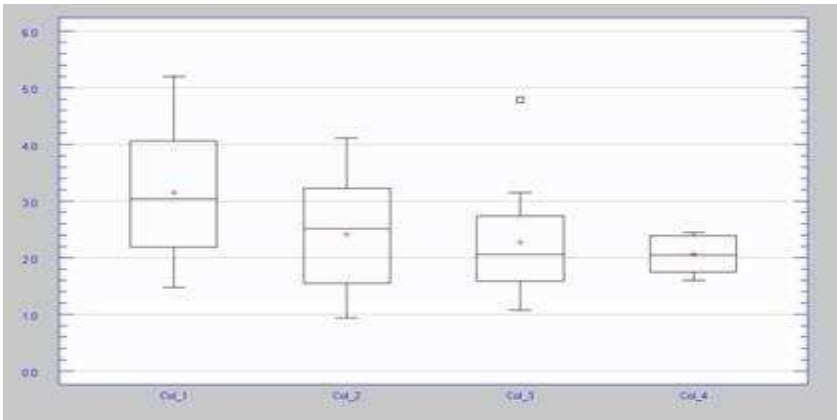


FIGURE 4.18: Response time to question 3 for different conditions.

items from the antecedent), there was no significant difference among the 4 metaphors.

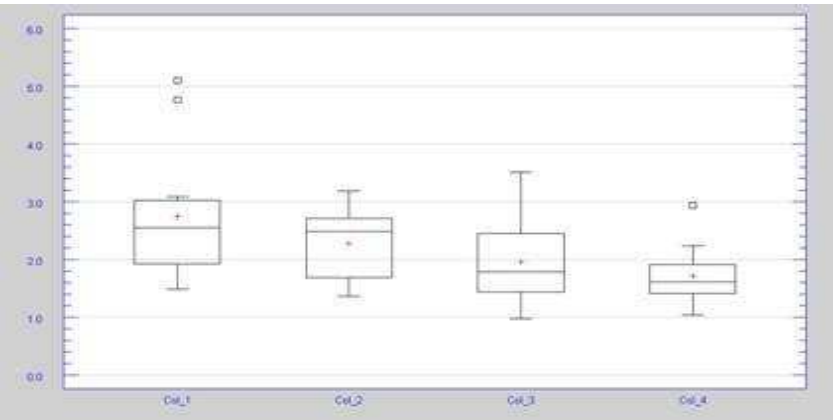


FIGURE 4.19: Response time to question 4 for different metaphors.

4.4.2.2 Error rate

For the error rate analysis, we consolidated the 5 questions together in the same histogram. Figure 4.21 shows that there is no differences among the metaphors. Each metaphor has its advantages and disadvantages. For instance, its is easier to find the most influential items (question 2) with metaphor 1 than with the other metaphors because the spheres are classified in ascending order. For question 3 (the most correlated items) it is easier the find the closest spheres with metaphors 3 and 4. For question 4 (the lowest correlated item) it the information is given only by metaphor 2.

Figure 4.22 shows that the use of Wiimote^{TM} produces a lower error rate relative to the mouse. When using the Wiimote^{TM} users take much more time to analyse the

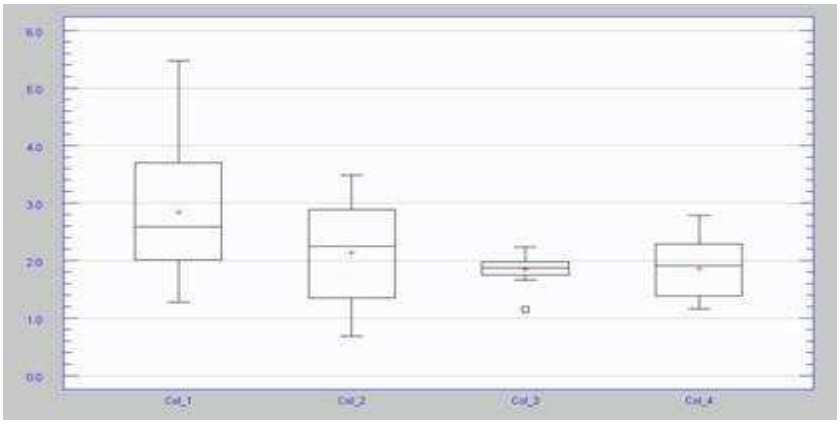


FIGURE 4.20: Response time to question 4 for different conditions.

representation than with the mouse and produce better results. This shows that the Wiimote^{TM} is more suitable in a large screen configuration. There is no differences between stereoscopic and monoscopic display.

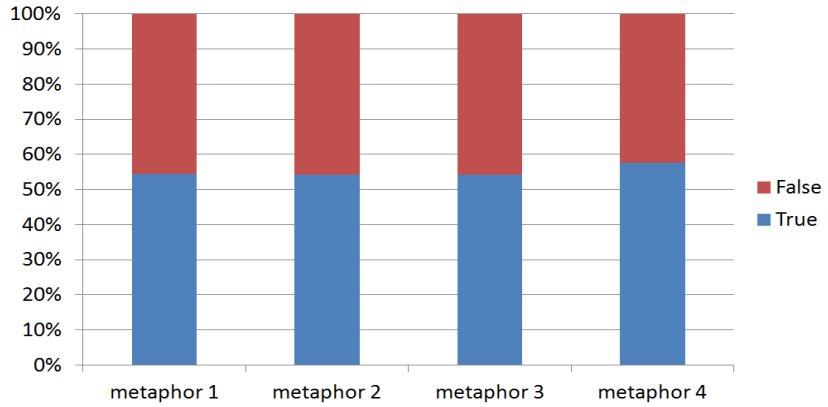


FIGURE 4.21: Error rates of the questions for different metaphors.

4.4.2.3 Subjective Aspects

The observation of the subjects performed during the study revealed that they have no difficulty to interact with the different representations of association rules. However, the majority of subjects (90 %) did not use the zoom function. This means that the distance from the display of the representation was adequate for all the metaphors and all the subjects. We also note that 70 % of the subjects preferred to use the Wiimote^{TM} compared to the traditional computer mouse. An adjustment period was necessary for the Wiimote^{TM} which explains the response time being still higher compared using the mouse. A more detailed experience concerning the use of the Wiimote^{TM} is necessary for more reliable conclusions to be drawn.

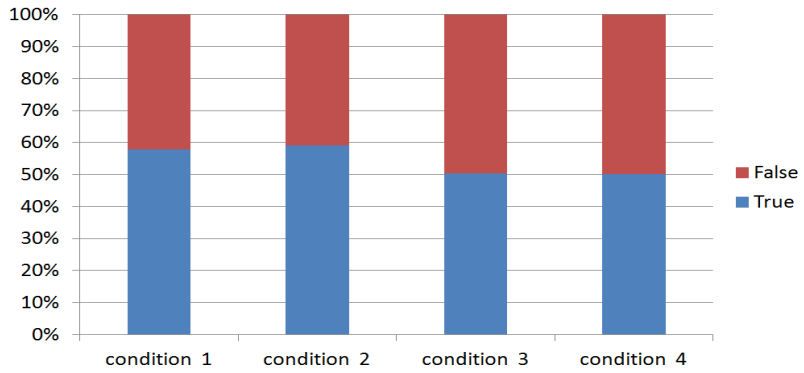


FIGURE 4.22: Error rates of the questions for different conditions.

4.4.3 Discussion

We have presented a series of experiments we conducted to study the relevance and the influence of different display configurations (monoscopic and stereoscopic) and interaction devices (mouse, wiimote^{TM}) in different tasks involving knowledge extraction from different representation association rule metaphors. Different experimental tasks using several interaction configurations have been proposed and validated. A first experiment was conducted to choose a metaphor among the four proposed to extract information about association rules with an acceptable compromise between response time and error rate. This experiment proves that metaphor 3 and metaphor 4 are the best in terms of response time. In a second step the choice of a configuration interaction/visualisation, the experimentation shows that there is no difference between the monoscopic display and stereoscopic display. In interaction, the best results were obtained with the mouse. The use of stereoscopy or monoscopy led to similar results.

The first experiment allowed us to choose a representation of association rules but did not conclude on the configuration interaction and visualisation to choose. The data collected through the questionnaire and observation of subjects during the experiment showed that wiimote^{TM} provided a sense of freedom of movement more than the mouse. 90 % of subjects preferred the wiimote^{TM} to a mouse. Some users mentioned recurring discomfort when viewing stereoscopically. Using a desktop PC with a monoscopic display to answer questions was annoying in the case of stereoscopic display. Some subjects were forced to remove their glasses which generated a longer response time for stereoscopic over the monoscopic viewing.

4.5 Interactive Visualisation of Association Rules with IUCEARVis

The IUCEAR methodology defines some basic principles for developing a tool dedicated to association rule exploration. However, the methodology can be implemented in multiple ways. In particular, various possibilities have been conceivable for graphic encoding. In this section we describe the choices that were made to work out the methodology for the visualisation tool.

To implement the four parts of the IUCEAR methodology, three different interfaces were presented to the user (Figure 4.23) for the:

- selection of interesting items;
- exploration and validation of recommended association rules;
- visualisation of the history map of the association rules selected by the user.

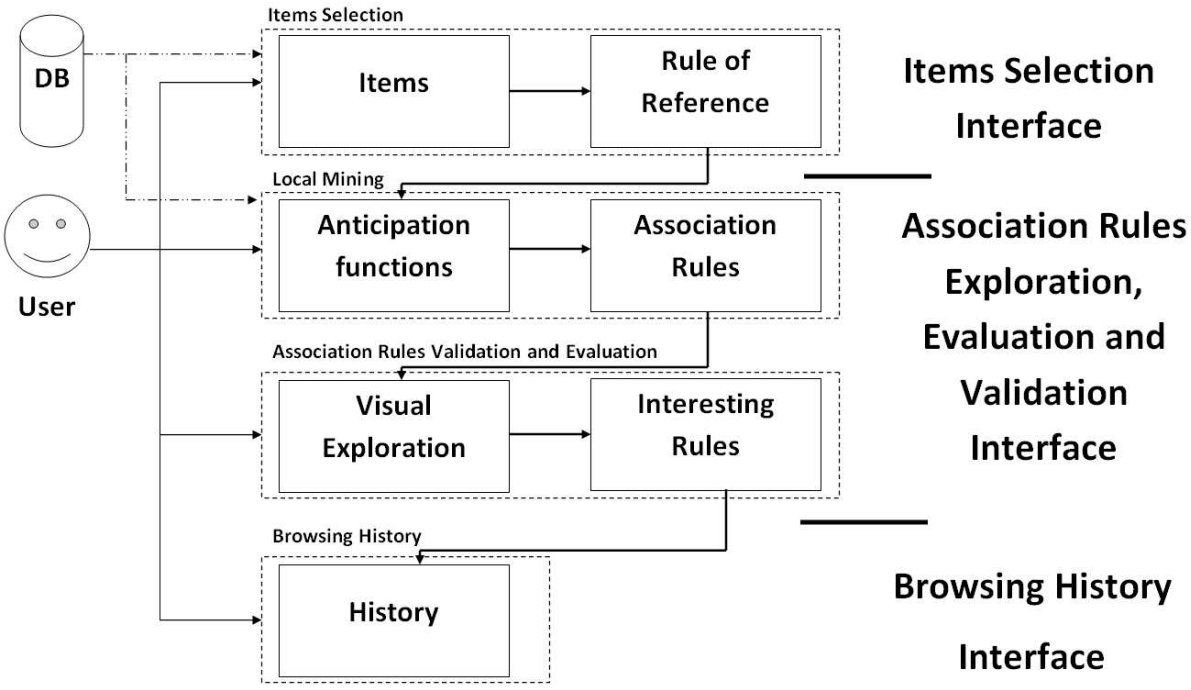


FIGURE 4.23: Illustration of IUCEARVis approach.

As we have seen in Section 4.4, the association rule metaphor was designed to display the items making up the rules. This metaphor is used in IUCEARVis to implement the IUCARE methodology. Such is the association rule metaphor, that we

chose an encoding approach based on positions and sizes to highlight the degree of rules interestingness in IUCEARVis.

We pursue the visualisation process proposed by Card *et al.* 1999 [60] (presented in Section 3.2.2) to describe the different visualisation interfaces.

4.5.1 Items Selection

4.5.1.1 Data Transformations

We took advantage of the fact that the user focuses on rule subsets to display only the subset being explored (cognitive principle (P1) of the IUCEAR methodology).

Initially, the user selects the interesting items that should be in the extracted association rules. Therefore, the user should build a *reference rule* by selecting the items that will make up its antecedent and the items that will make up its consequent. Thus, the functions *add to antecedent* and *add to consequent* perform the data transformation from a set of items I to a rule A. We call A the *reference rule*, because it is from the component items of this rule that rules will be generated.

For example, with a set of four items I = (A,B,C,D) the user can build any rule that he considers interesting (milk $\rightarrow$ bread) ; (milk, bread $\rightarrow$ apples), (milk, bread, apples $\rightarrow$ eggs), etc.

To start or restart a rule exploration process, the user has first to select a reference rule using the item selection interface.

Example 4.5.1 In the example shown in Figure 4.24, the user builds the reference rule (con=+, n=+, led=+ $\rightarrow$ clv=0). He/she can immediately note that this is a poor quality rule (confidence = 27%, support = 4%, and lift = 0.57).

4.5.1.2 Rendering Mappings

Multiple objects are present in this interface (Figure 4.25):

1. list of items: a pick-list contains all database items;

2. cylinders representing the frequency of items: the maximum size of a cylinder = 1. The most common item in the database will be represented by a cylinder of size 1. The cylinders may have different colours: blue, green or red. More precisely, when there is more than one item in the antecedent and more than one item in the consequent, the contribution of the item to the association rule can be calculated if the item will be added to the antecedent or if the item will be added to the consequent. The cylinder colour indicates if the item improves the quality of the rule's interestingness. A green colour indicates that the item

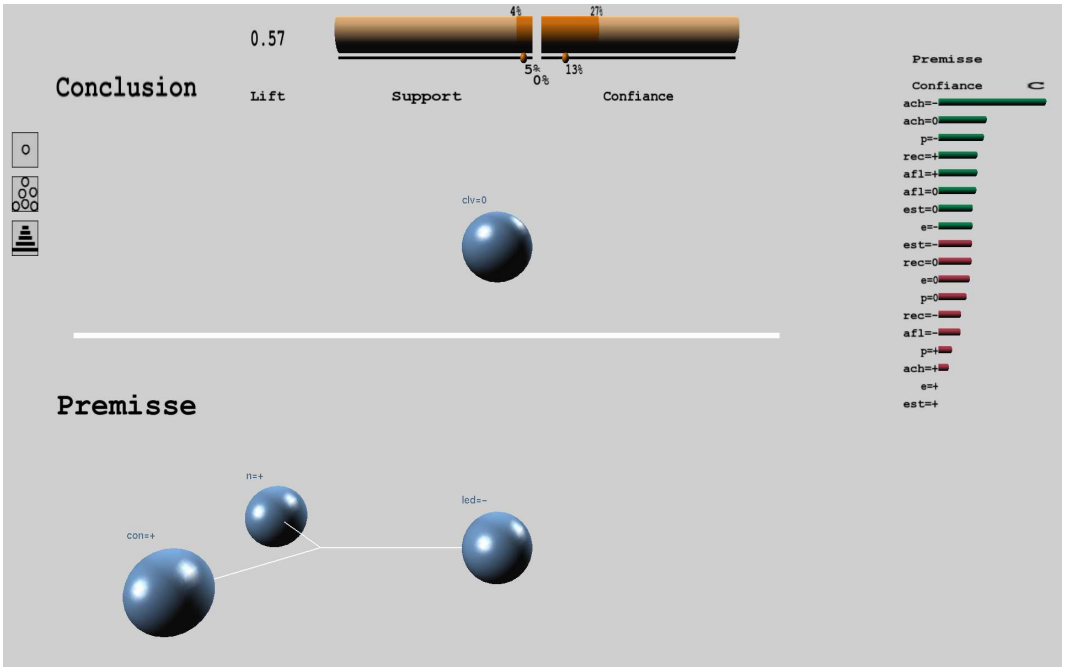


FIGURE 4.24: Item Selection interface.

improves the rule quality. A blue colour indicates that the item does not change the rule quality. A red colour indicates that the item degrades the rule quality.;

3. spheres representing the antecedent items have a fixed size. Consequently, the spheres are all identical;

4. spheres representing the consequent items are identical to the spheres representing the antecedent items;

5. cylinders representing the support and confidence they are the same size. The maximum size of the cylinder is equal to 100 % of the interestingness measure. To represent the values of the reference rule interestingness measures we chose a graphic encoding based on sizes: the size of a cylinder. The value is represented by the darker colour, and is also displayed above the cylinder. Since the lift is an unbounded measure we decided to display only the value without a graphic object associated to it;

6. a cursor is used to select the thresholds values of interestingness measure (support and confidence). The threshold values may vary between 0% and 100%.

In this interface we do not use the metaphor of representation of association rules presented in Section 4.4. This metaphor allows the comparison of association rules, whereas in this interface only one rule is displayed: the reference rule.

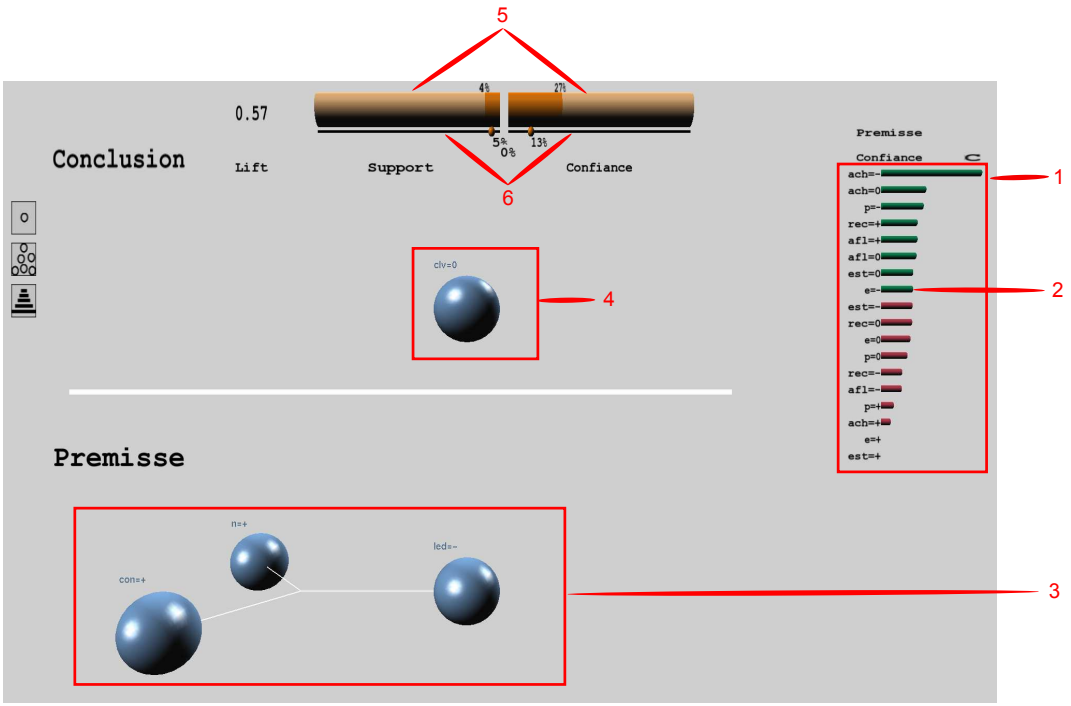


FIGURE 4.25: Objects present is the item Selection interface.

4.5.1.3 View Transformation

This interface allows the user to build a reference rule and display the interestingness measure related to it. Firstly, the user visualise all items (I) present in the database in a pick-list. The number of items displayed on the drop down is set to 28. If there are more then 28 items in the database, the user can easily scroll down the list to show the hidden items. In order to help the user to get an idea about the data and to choose interesting items, the item selection interface offers the possibility to modify the ordering items list criteria. The item pick-list can be ordered in frequency order, alphabetical order, highest support, highest confidence, and highest lift (the ordering by interestingness measures is possible only if at least one item occurs in the antecedent and one item in the consequence). The user can choose to visualise the contribution of the item to the association rule if it will be in the antecedent or if it will be in the consequence. The user can also change the support and confidence minimum thresholds by the algorithms of rule extraction.

To navigate among the different interfaces (item selection, rule exploration, and browsing history) a command menu allows direct access to the desired interface. This command menu is present in the three interfaces of IUCEARVis.

4.5.2 Association Rule Exploration, Evaluation and Validation

After item selection, the system proposes to the user a small set of rules syntactically close to the reference rule. Then, the user can visualise, evaluate and validate them through the association rule exploration interface. The algorithms used for association rule generation will be presented in Section 5.2

4.5.2.1 Data Transformation

After selecting a reference rule A, the anticipation functions perform transformation of a rule A to a set of rules R by targeting subsets of rules.

Six anticipation functions are taken up in IUCEARVis. Most of them are specialisation functions or generalisation functions. Specialisation and generalisation are indeed the two fundamental cognitive processes to generate new rules according to Holland *et al.* 1986 [147]. One of the characteristics of the anticipation functions is the integration of the most important interestingness measures of association rules, support ($supp$) and confidence ($conf$). Each interestingness measure is associated to a minimum threshold ($minsupp$ and $minconf$) set by the user to filter rules. These thresholds can be changed by the user at any time during navigation in R.

To define the anticipation functions, we bring thresholds together in the boolean function *GoodQuality*:

$$\forall r \in R, GoodQuality(r) \Leftrightarrow (supp > minsuppandconf > minconf)$$

This approach is in the form of the following anticipation functions:

- The association rule of reference has only one item in the antecedent or only one item in the consequent

- $A_1(x \rightarrow Y) = z \rightarrow Y | z \in I \ (Y)$
- $A_2(X \rightarrow y) = X \rightarrow z | z \in I \ (X)$

- The association rule of reference has at least one item in antecedent and one item in the consequent

- anticipation functions of specialisation: specialisation consists of adding an item to the rule antecedent or the rule consequent;

  – $A_3(X \rightarrow Y) = X \cup z \rightarrow Y | z \in I \ (X \cup Y) \wedge GoodQuality(X \cup z \rightarrow Y)$

– $A_4(X \to Y) = X \to Y \cup z | z \in I\ (X \cup Y) \wedge GoodQuality(X \to Y \cup z)$

- anticipation function of generalisation: generalisation consists of simplifying the rule antecedent or the rule consequent.

– $A_5(X \to Y) = Xz \to Y | z \in X \wedge GoodQuality(X\ z \to Y)$
– $A_6(X \to Y) = X \to Yz | z \in Y \wedge GoodQuality(X \to Y\ z)$

For the purpose of simplicity, upper case (X, Y) denotes item sets and lower case (x, y, z) denotes items. We note $X \cup Y$ instead of $X \cup Y$ and $X \backslash z$ instead of $X \backslash z$.

Example 4.5.2 Let us consider a sample of supermarket transaction data set presented in Table 4.5.2 with $minsupp = 20\%$ and $minconf = 80\%$:

Tuple	Milk	Bread	Eggs	Apples	Pears
1	1	0	1	0	1
2	1	1	0	1	1
3	1	1	1	1	0
4	1	1	1	1	1
5	0	0	1	1	0

TABLE 4.2: A supermarket transaction data set.

Let us consider an association rule

$$Milk, Bread \to Apples\ [supp = 60\%\ and\ conf = 100\%]$$

- if we used A_2, two association rules can be generated (Milk, Bread → Eggs [Supp= 40% and Conf= 66%]) ; (Milk, Bread → Pears [supp=40% and conf=66%]). All the rules have $conf \leq minconf$;
- if we used A_3, two association rules can be generated (Milk , Bread, Eggs → Apples [supp= 40% and conf=100%]) ; (Milk, Bread, Pears → Apples [supp=40% and conf=100%]). The two rules have $supp \geq minsupp$ and $conf \geq minconf$;
- if we used A_4, two association rules can be generated (Milk, Bread → Apples, Eggs [supp=40% and conf=66%]) ; (Milk, Bread → Apples, Pears [supp=20% and conf=50%]). All the rules have $conf \leq minconf$;
- if we used A_5, two association rules can be generated (Milk → Apples [Supp=60% and Conf=100%]) ; (Bread →, Apples [Supp=60% and Conf=100%]). The two rules have $supp \geq minsupp$ and $conf \geq minconf$;

- A_1 and A_6 can't be used because there is a single item in the antecedent.

A total of 8 association rules can be generated, but only 4 will be presented to the user ($supp \geq minsupp$ and $conf \geq minconf$).

- *R1: Milk → Apples :Supp(R1)=60% and Conf(R1)=100%*;
- *R2: Bread → Apples Supp(R2)=60% and Conf(R2)=100%*;
- *R3: Milk, Bread, Eggs → Apples Supp(R3)= 40% and Conf(R3)=100%*;
- *R4: Milk, Bread, Pears → Apples Supp(R4)=40% and Conf(R4)=100%.*

According to the number of items present in the database, the number of generated association rules can be very high. As we have already seen in Chapter 1, human capabilities can't deal with large volumes of information. To limit the cognitive load of analysis, we propose a limited subset of rules at a time. Only the 10 most important rules will be displayed. In order to select those rules, the user must choose one of the selection criteria among :

- highest support;
- highest confidence;
- highest lift;
- an aggregate measure = weightSupport * support + weighconfidence*confidence + weightlift* lift weights values are defined by the user.

The algorithms developed for the local extraction of association rules are presented in Section 5.2.

4.5.2.2 Rendering Mappings

Association rule exploration, evaluation and validation is the central interface of IUCEARVis. It is through this interface that the user explores the subset of rules generated by the local mining algorithms.

Based on the *reference rule* A, the system generates and displays a subset of rules $S = F(A)$ that may interest the user. The *reference rule* will be consistently added to any generated subset. This allows comparisons between the *reference rule* and the rules proposed by the system. For example, it is interesting to identify which items can be removed from the rule without degrading the rule quality. Reciprocally, comparison can be used to check if adding new items to the antecedent or to the consequent improves the rule quality.

The association rule exploration interface presents 3 different regions: (Figure 4.26) :

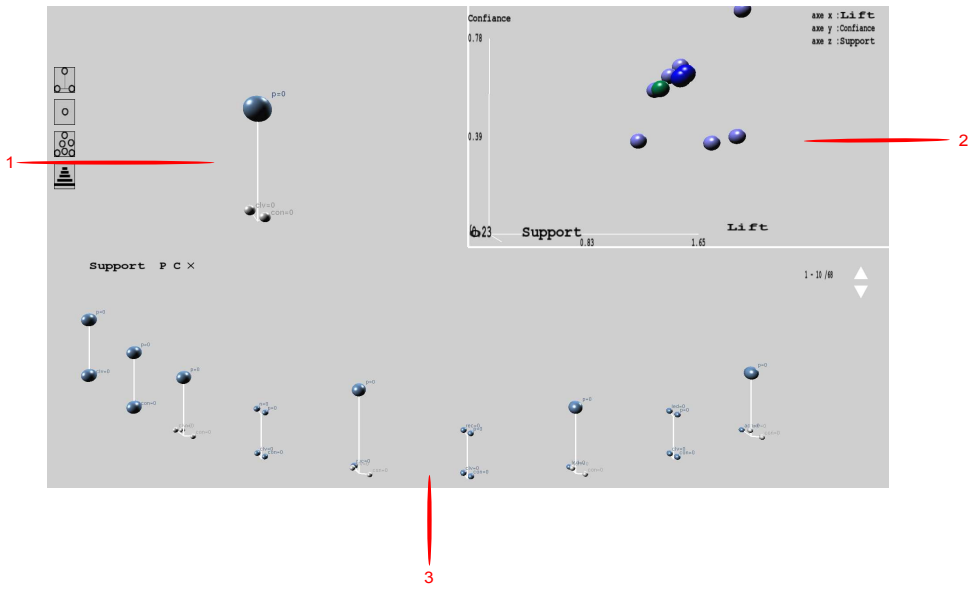


FIGURE 4.26: Interface for association rules exploration, validation, and evaluation.

1. at the upper left: the *reference rule* built by the user via the item selection interface. Displaying the rule of reference allows the user to compare the rule that he/she built with the rules proposed by the system. The *reference rule* is represented according to our rule metaphor;

2. at the upper right: a 3D Scatter plot (see description Chapter 3 Section 3.3.1.1) in which the rules are represented . As advocated by the IUCARE methodology, we chose map encoding based on positions to highlight the rule interestingness measures. In a 3D scatter plot, the position of an object might allow us to encode three rule interestingness measures, an interestingness measure on each axis. To facilitate the perception of depth (z axis) in the 3D space, the intensity of colour is associated with the depth axis. The closer the object is to the user, the more intense the colour. To distinguish the rule of reference, it will be represented in green while other rules are represented in blue. The scatter plot stresses the good quality rules. More precisely, a rule placed in the top of the three axes represents a rule whose support, confidence, and lift are high. On the other hand, a rule placed near the scatter plot origin represents a rule whose three interestingness measures are weak;

3. at the bottom: In a visual representation, the perceptually dominant information is the spatial position. Therefore the interestingness measures which are fundamental for decision making are represented by the object position. The user chooses what rules interestingness measure will serve to organise the rules. Since the range of *Information Gain* values vary according to the displayed rule set, IUCAREVis includes a normalisation procedure for allocating sphere sizes on larger values. This also helps to avoid low values lead to very small spheres

and large values lead to large spheres.

An association rule can have a score between 0 to 3 given by the user according to his/her interest to the rule. The colour of links in the representation metaphor is used to encode the score value as follows (Figure 4.27):

- 0: white colour;
- 1: azure colour;
- 2: medium blue colour;
- 3: dark blue colour.

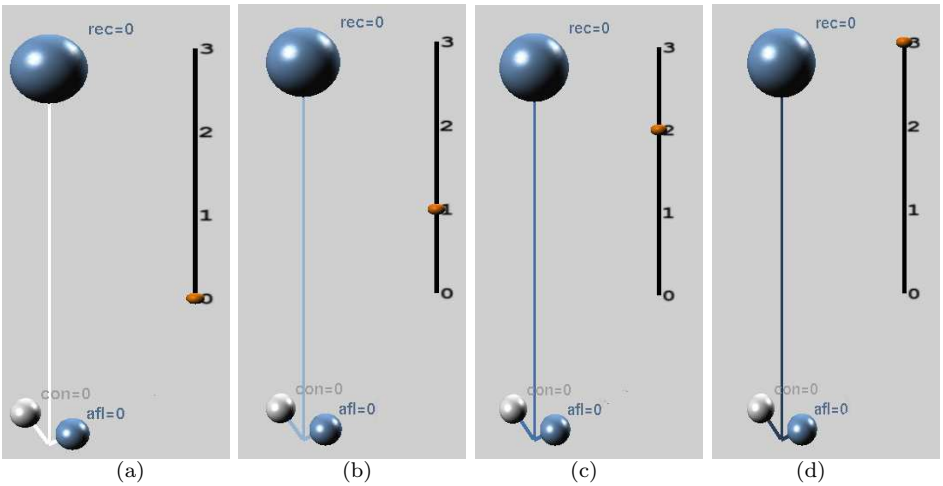


FIGURE 4.27: The different colours of links to encode rules score: (a): score 0 (white colour), (b): score 1 (azure colour), (c): score 2 (medium blue colour), (d): score 3 (dark blue colour).

4.5.2.3 View Transformation

Research work in visual perception shows that humans have at first overall perceptions of a scene, before paying attention to detail (Hascoet and Beaudoin-Lafon 2001 [139]), this motivated the development of *Focus/context* approaches (see Chapter 3 Section 3.2.2). This is especially well known in a formula made by Shneiderman 1996 [244] widely used in information visualisation: "overview first, zoom, and filter, then details on demand". In IUCAREVis, the user must be able to pass easily from a global view to a detailed view by interacting with the visualisation. For that, the user can focus on one rule among the displayed association rules set. The user selects the rule that he/she want to show and this rule will be displayed on the front of the screen. The other rules remain displayed but blurred. The selected rule is displayed in red to keep the overall context (Figure 4.28). The sphere representing the same

rule in the 3D scatter plot is displayed in red too. This technique is called brushing and linking (see Chapter 3 Section 3.2.2). The interestingness measure values relative to the selected rule are displayed. It is easy to locate the best rules with the scatter plot. The user can just click on the selected rule to zoom in and examine it more closely.

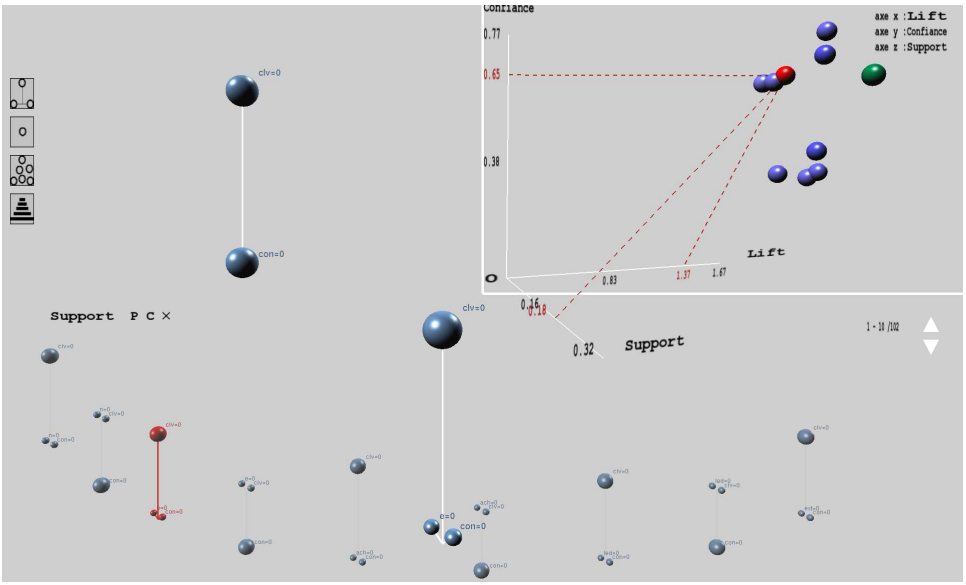


FIGURE 4.28: Linking and brushing: a selected rule is simultaneously highlighted in the 3D scatter plot.

Several system control commands are also available to meet the different needs of the user in his search for interesting knowledge (Figure 4.29):

1. set a new rule as a *reference rule*. The user can select any rule from the displayed set of rules and make it the new *reference rule*. The system automatically triggers the algorithms to generate a new subset of rules according to the anticipation functions.

2. modify the 3D scatter plot axis. To give more flexibility to the user, we give him/her the possibility to decide which interestingness measure will be represented on each axis.

3. change the displayed subset of rules. The association rules are displayed by a set of 10. The user can then display each time the previous or following subset.

4. change the rule interestingness measure used to sort the rules set. The user can may choose between : support, confidence, lift or global measure (see section 4.5);

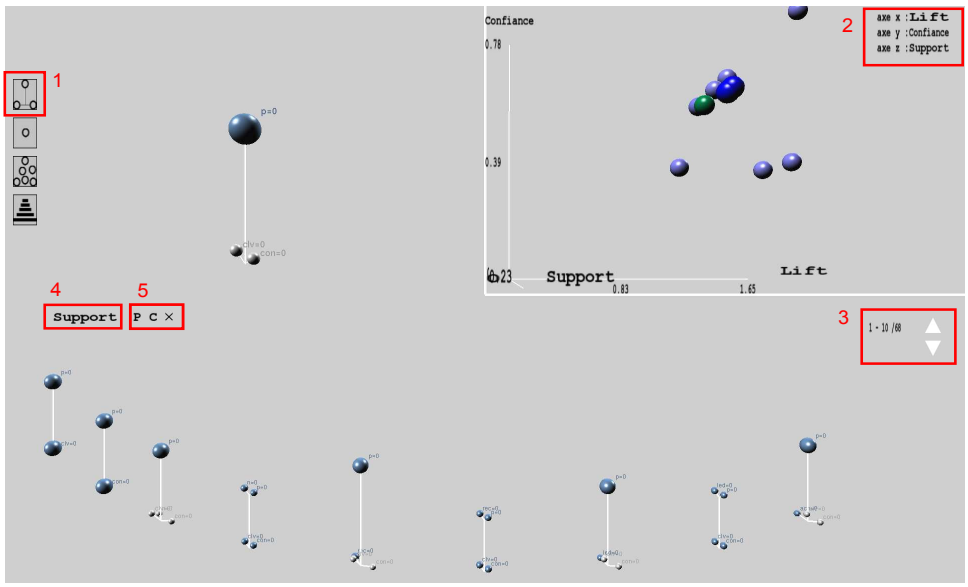


FIGURE 4.29: System control commands available in the rules exploration, evaluation and validation interface.

5. activate/deactivate filter. The user can visualise only rules that have the same consequent or the same antecedent as the reference rule;

6. set a rule score. During the exploration step, an interesting rule can draw the attention of the user, then, he/she can give it a score between 1 and 3 and add it to the history map of interesting rules. The cursor used to set the rule score is displayed at the user request (Figure 4.30).

4.5.3 Browsing History

Through the browsing history interface, the user visualises all rules which have drawn his/her attention during the exploration and extraction process. So there is no transformation of data.

4.5.3.1 Rendering Mappings

Unlike the exploration interface where the number of rules is limited to 10, this interface must support large numbers of rules. Therefore, we have chosen a representation based on the information landscape metaphor to place the association rule set. Specifically in a scale (Figure 4.31).

Since in a visual representation the perceptually dominant information is the spatial position, the interestingness measures which are fundamental for decision making are represented by the association rule position in the scale. In a 3D information landscape, the position of an object allow us to encode three interestingness measures.

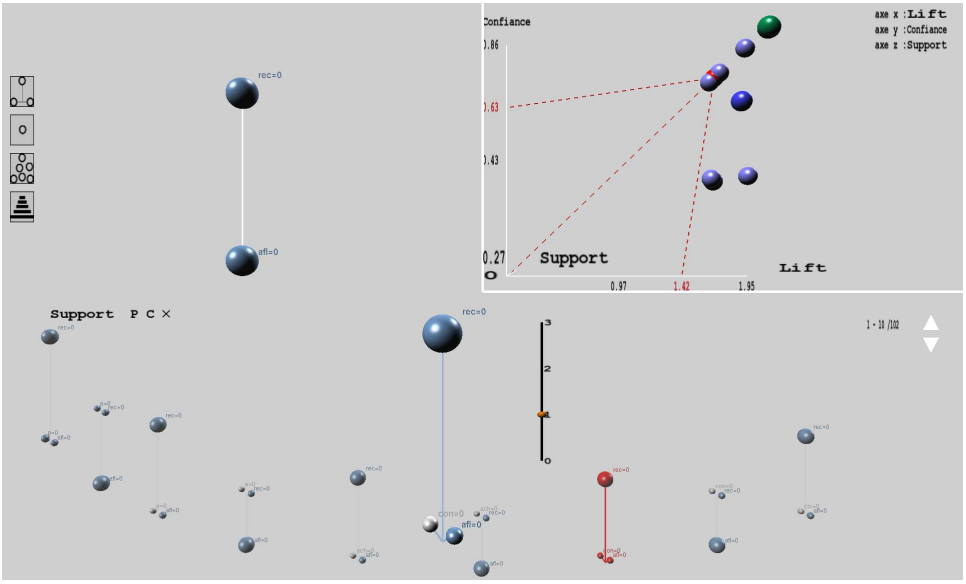


FIGURE 4.30: A cursor can be displayed at the user request to change a rule note.

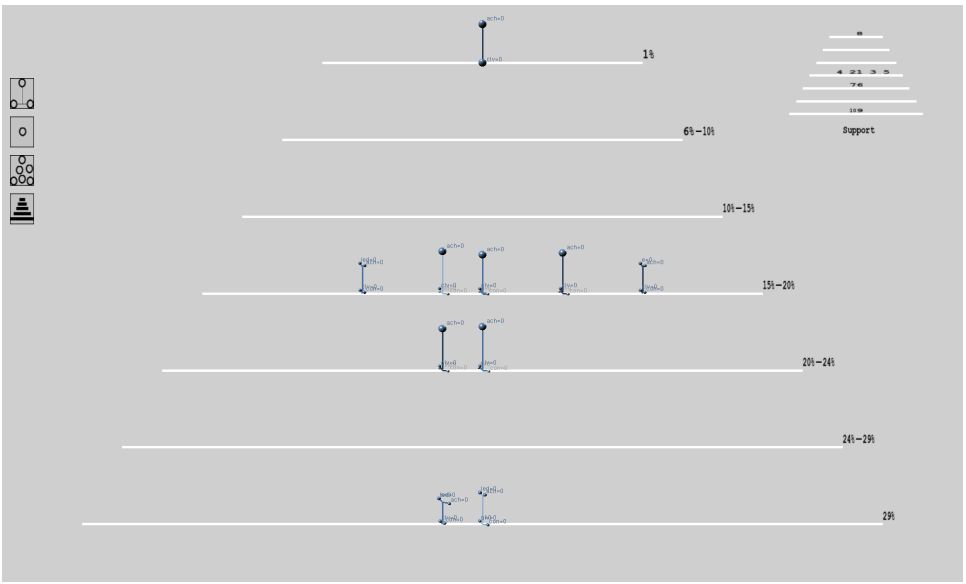


FIGURE 4.31: Interface for browsing history.

As our metaphor for association rule representations uses sphere size to represent the *informational gain* items and in order to ensure that the user is not skewed, we chose not to use depth to place rules. A sphere placed at depth may seem smaller than a sphere with the same size placed in front. So a user can not compare correctly two rules placed at different distances from him/her.

However, given that different rules may have the same interestingness measures

values, it is necessary to leave one dimension in order to distribute the association rules in space and prevent them from overlapping. Only one interestingness measure is encoded by the position. In addition, the rules placement on a scale allow highlighting the rules with good quality. More precisely, a rule positioned on the bottom step is a rule whose interestingness measure value is greater than another placed on the top step. A landscape miniature representation allows users to have the order of addition of each rule to the history. Instead of each rule displayed in the scale, a number corresponding to the order of addition of the rule is displayed in the miniature scale.

The scale is composed of 7 steps and on each step rules are distributed from the centre of the step towards the outside. The size of the scale is the same regardless the number of displayed rules. However, interestingness measure values corresponding to each step vary depending on the displayed rules (Figure 4.32). Maximum and minimum interestingness measure values are calculated each time a rule is added or deleted from the displayed rule set.

For the association rule representation we use our metaphor. The colour of links among the items corresponds to the note that the user has given to the rule like in the exploration interface. An *Information Gain* normalisation procedure is used normalise the sphere sizes.

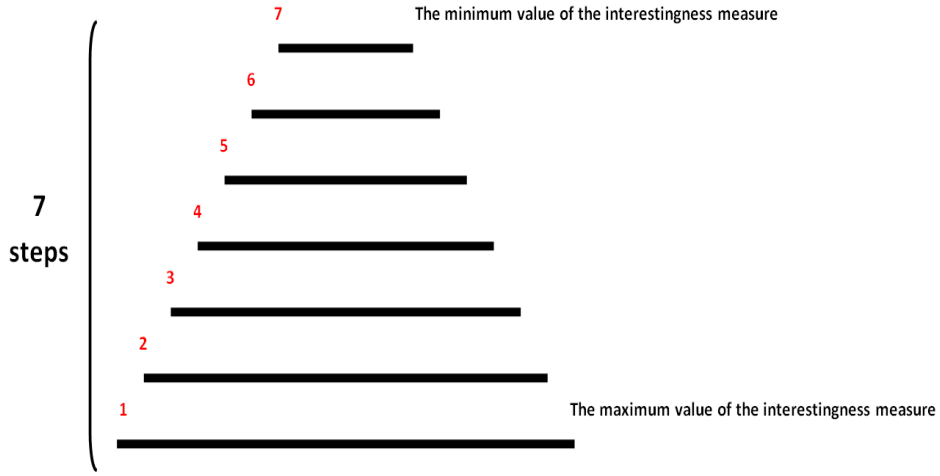


FIGURE 4.32: The rule positions on the scale are based on the interestingness measure values.

4.5.3.2 View Transformation

As in the rule exploration and validation interface, the user can focus on one rule among the displayed association rule set. The user selects the rule that he/she wants to show and this rule will be displayed on the front of the screen. The other rules remain displayed but blurred. The selected rule is displayed in red to keep the overall context. The user can select rules from the miniature scale too (brushing and linking technique see Chapter 3 Section 3.2.2).

The user can modify the score of a given association rule at any time. Then the link colour will be modified directly. The user can also delete an association rule from the history by giving it the score 0. It will immediately be removed from the interface.

4.6 Conclusion

In this chapter, we have presented the IUCEAR methodology for interactive visualisation of association rules and IUCEARVis an implementation of the IUCEAR methodology.

IUCEAR methodology is designed for the convenience of the user faced with large sets of rules taking into account his/her capacity of information processing. We propose also a new association rule metaphor allowing the visualisation of attributes composing the association rule. Also, it shows attribute relationships and the contribution of each one of them.

IUCEARVis allows rule exploration and the identification of relevant knowledge. The tool is based on an intuitive VR representation that supports the extraction of sets of rules described by several interestingness measures. This representation highlights rules interestingness measures and facilitates the re cognition of good quality rules. IUCEARVis proposes anticipation functions (generalisation and specialisation) to help the user find interesting rules. Many interaction operators allows users to guide navigation among rules.

As shown in Chapter 1 Section 1.5.3, 2D association rule visualisation tools are based on items and see the rule interestingness measures as ancillary information. On the other hand, a few association rule visualisation tools (Chapter 3 Section 3.4.2.3) are proposed in 3D and VR. Those tools propose: either representations mainly based on rule interestingness measures (Blanchard *et al.* 2007 [31]), or a representation mainly based on items but without highlighting rule interestingness measures (Gotzelmann *et al.* 2007 [130]). Other proposed methods offer very few interaction operators. IUCEARVis is quite different for two mains reasons because it proposes (i) representation based on both rule items and interestingness measures, and (ii) many interaction operators based on both rule items and interestingness measures.

5 IUCEARVis Tool Development

Contents

5.1 Introduction

Merely developing a novel visual metaphor is rarely sufficient to make new discovery. In the association rule extraction process, the decision-maker is overwhelmed by the association rule algorithm results. Representing these results as static images limits the use of the visualisation. This explains why the user needs to be able to interact with the association rules representation in order to find relevant knowledge.

Interaction also allows the user to be integrated into the association rule extraction process. The user should be able to manipulate the extraction rule algorithms and not only the graphical representations. This allows him to focus on interesting knowledge from his point of view, in order to make the association rule methods be

more generically useful.

In this chapter we firstly present the IUCEARVis rule extraction module. Then we detail its implementation. Rule extraction is an interactive module that generates sets of rules and their interestingness measures at each user demand. Thus no rule is produced in advance. This local extraction approach follows user exploration. It is conceived to overcome the limitations of exhaustive extraction with which the time taken may be prohibitive for dense data (Chapter 1 Section 1.5). The local extraction of association rules is carried out by constraint-based algorithms, designed specifically to extract on demand the sets of rules generated by the anticipation functions.

In this chapter, we are also interested in presenting the implementation of the IUCEARVis tool. The tool is based on virtual reality visualisation and intuitive 3D interaction techniques. Firstly, we present the local association rule extraction algorithms. Then we detail the implementation of IUCEARVis in Section 2. Section 3 is devoted to interaction techniques in IUCEARVis. Finally, we present a case study.

5.2 Interactive Rule Local Mining With IUCEARVis

Many algorithms for association rule extraction exist in the literature (see Chapter 1 Section 1.4). One of the most popular is the *Apriori* algorithm. *Apriori* uses an incremental approach to find all frequent item sets – that have support above the minimum threshold. Then, it builds all rules that have confidence above the minimum threshold. The problem with this algorithm is the large amount of generated association rules which makes their analysis almost impossible. Another approach consists of searching for interesting rules locally, in the neighbourhood of rules that the user already knows. Instead of generating all rules by means of an exhaustive algorithm, the new approach consists of generating locally all candidate rules (syntactically close to the rule selected by the user), and then checking their support and confidence against the transaction database. This local approach follows user navigation and overcomes the limitations of exhaustive extraction algorithms. The candidate rules are all possible interesting rules. After generation, a pass over the database is performed to compute rule interestingness measures. In order to be present in the output rules set, rules must conform with the support and confidence thresholds specified by the user. This association rule extraction module is interactive and generates sets of association rules on demand. Local rules extraction is carried out by a constraint-based algorithm, specifically designed to extract rule sets at the user's request.

The IUCEARVis constraint-based algorithms were designed with a quite different perspective to the constraint-based algorithms presented in Chapter 1, Section 1.4. Whereas the latter is looking to exploit constraint classes as generally as possible, the IUCEARVis algorithm uses only special constraints induced by the anticipation functions.

5.2.1 Constraints in IUCEARVis

The constraint-based algorithms use constraints to reduce the search space. In IUCEARVis, there are two types of constraints on rules.

- syntactic constraints: what items can or must appear in the antecedent and the consequent;
- quality constraints: that specify a minimum threshold for both rule interestingness measures of support and confidence.

The syntactic constraints are the most powerful constraints. They allow the search space to severely restricted.

5.2.2 Association Rule Extraction in IUCEARVis

The general local algorithm for association rules extraction is presented in Table 5.2.2.

This algorithm is organised into four main steps:

- **STEP 1**: in this step we use syntactic constraints to construct candidate association rules. This function enumerates all association rules that satisfy a syntactic constraint only from the items list without having to consult individuals in the database. Accordingly, this step does not require any pass over the database;
- **STEP 2**: the purpose of this step is to calculate the rule interestingness measure values. First, we must calculate the cardinality of each itemset – number of individuals who check the itemset in the database, of the rules listed in STEP 1. For the rule: (*Milk, Bread* $\rightarrow$ *Eggs*), three cardinals itemsets should be calculated, the antecedent cardinal ($n_{milk,bread}$), the consequent cardinal (n_{eggs}), and the global itemset [antecedent $\cup$ consequent] ($n_{milk,bread,eggs}$). The cardinals are determined by counting the itemset occurrences in the database. This step is the most time expensive step. Once the cardinals are determined, the rule interestingness measure values can be calculated. The number of transactions in the database (n) is also necessary to calculate the rule interestingness measures;
- **STEP 3**: in this step the quality constraints are exploited. The rules are filtered on the rule interestingness measures relative to the support threshold and confidence threshold.
- **STEP 4**: in this step the rules are ranked according to an interestingness measure specified by the user. Default configuration uses the support; otherwise, the last ranking measure chose by the user.

Input: Database D
Association Rule (Reference Rule) RR
Set of candidates Items I
Thresholds (minimum Thresholds of rules interestingness measures)
$minSupp$ and $minConf$
Ranking interestingness measure RM
Output: Set of couple (AR, M(AR))
when AR is an association rule and M(AR) its interestingness measures $SetAR$

1. $SetAR = \emptyset$
STEP 1: Construction of candidate rules with syntactic constraints
2. **forall** (I_k in I) **do begin**
3. AR = construction-rule(I_k, RR)
4. **endfor**

STEP 2: Calculate the association rule interestingness measure
5. **forall** (AR_k in AR) **do begin**
6. $M(AR)$ = calculate-interestingnessMeasures(AR_k, D)
7. **endfor**

STEP 3: Eliminate the candidate rules that do not meet the quality constraints (support threshold and confidence threshold)
8. **forall** (AR_k in AR) **do begin**
9. $SetAR$ = filters(AR_k,$M(AR)$, $minSupp$,$minConf$)
10. **endfor**

STEP 4: Rank the candidate rule set according to the interestingness measure specified by the user
11. $SetAR$ = Ranking($SetAR$,RM)

12. **return** $SetAR$

TABLE 5.1: The local association rule extraction algorithm.

In the following, we give a more detailed description of the local algorithms. Each algorithm corresponds to one of the anticipation functions of the IUCEAR methodology.

The algorithm presented in Table 5.2.2 is used to extract sets of rules defined by the specialisation anticipation function. This algorithm is used only if there is more than one item in the antecedent or more than one item in the consequent. Firstly, in line 2, the algorithm enumerates all candidate items I_k. I_k should not belong

to the antecedent or to the consequent of the reference rule (line 3). Secondly, the algorithm creates a new candidate rule by adding the new item to the antecedent of the reference rule ($PRR \cup I_k \rightarrow CRR$) (line 4). In lines 5, 6, and 7 the algorithm determines the cardinals of the itemsets needed to calculate the rule interestingness measures (support, confidence, and lift). If the new rule interestingness measure values are above the thresholds (line 11), the new rule can be added to the result set (line 12). The rules that do not satisfy the support and the confidence thresholds are eliminated. Thirdly, the algorithm creates a new candidate rule by adding the new item to the consequent of the reference rule ($PRR \rightarrow CRR \cup I_k$) (line 13). The same treatment will be repeated for this new rule (lines 14 to 18).

If there is only one item in the antecedent or one item in the consequent of the reference rule, a specific specialisation algorithm is used to generate association rules. This algorithm is presented in Table 5.3.

The algorithm presented in Table 5.4 is used to extract sets of rules defined by the generalisation anticipation function. This step generates a very small number of association rules. The algorithm involves two important steps. Firstly, for rules that the number of items in the antecedent is greater than one item, the algorithm eliminates at each iteration one item. The new candidate rule is of the form: $AR_i = (PRR \ \ I_k \rightarrow CRR)$. Then the algorithm calculates this rule interestingness measure and adds it to the rule list if their interestingness measure values satisfy the thresholds.

5.3 Implementation

5.3.1 Virtual Reality Technology

In IUCEARVis, we used OpenGL to generate 3D scenes. OpenGL serves two main purposes: (i) it presents a single, uniform interface for different 3D accelerator hardware and (ii) it supports the full OpenGL feature set, using software emulation if necessary, for all implementations. OpenGL is evolutionary; it allows additional functionality through extensions as new technology is created. Several libraries are built on top of or beside OpenGL to provide features not available in OpenGL itself. Libraries such as GLU can be found with most OpenGL implementations, and others such as GLUT and SDL have grown over time and provide rudimentary cross-platform windowing and mouse functionality. OpenGL does not load directly the display. It only describes 3D objects, initialises OpenGL rendering which is done by the operating system API or by the OpenGL Utility Toolkit, GLUT, which is a window system independent toolkit for writing OpenGL programs. It implements a simple windowing application programming interface (API) for OpenGL. The comparison between 2D, 3D and virtual reality approaches realised in Chapter 2, Section 2.4.2 allowing us to believe that only the stereoscopic enables fully and efficiency exploitation of 3D representations.

Input: Database D
Antecedent of the Reference Rule PRR
Consequence of the Reference Rule CRR
Set of candidates Items I
Minimum support Thresholds $minSupp$
Minimum confidence Thresholds $minConf$
Ranking interestingness measure RM
Output: a set of couple (AR, M(AR)) when AR is an association rule and M(AR) its interestingness measures $SetAR$

1. $SetAR = \emptyset$
2. **forall** (I_k in I) **do begin**
3. **if** $I_k \notin PRR$ and $I_k \notin CRR$ **then**

STEP 1
4. $AR_i = PRR \cup I_k \rightarrow CRR$

STEP 2
5. calculateCardinal($I_k \cup PRR$,D)
6. calculateCardinal($I_k \cup CRR$,D)
7. calculateCardinal($I_k \cup CRR \cup PRR$,D)
8. $M(AR_i, support)$ = calculateSupport(AR_i)
9. $M(AR_i, confidence)$ = calculateConfidence(AR_i)
10. $M(AR_i, lift)$ = calculateLift(AR_i)

STEP 3
11. **if** $support(M(AR_i)) \geq minSupp$ and $confidence(M(AR_i)) \geq minConf$ **then**
12. $SetAR = SetAR \cup (AR_i, M(AR_i))$

STEP 1
13. $AR_{i+1} = PRR \rightarrow CRR \cup I_k$

STEP 2
14. $M(AR_{i+1})(support)$ = calculateSupport(AR_i)
15. $M(AR_{i+1})$ $(confidence)$ = calculateConfidence(AR_{i+1})
16. $M(AR_{i+1})(lift)$ = calculateLift(AR_{i+1})

STEP 3
17. **if** $support(M(AR_{i+1})) \geq minSupp$ and $confidence(M(AR_{i+1})) \geq minConf$ **then**
18. $SetAR = SetAR \cup (AR_{i+1}, M(AR_{i+1}))$

19. **endfor**

STEP 4
20. $SetAR$ = Ranking($SetAR$,RM)

21. **return** $SetAR$

TABLE 5.2: The local specialisation anticipation function algorithm.

Input: Database D
Antecedent of the Reference Rule PRR
Consequence of the Reference Rule CRR
Set of candidates Items I
Minimum support Threshold $minSupp$
Minimum confidence Threshold $minConf$
Ranking interestingness measure RM
Output: a set of couple (AR, M(AR)) when AR is an association rule and M(AR) its interestingness measures $SetAR$

1. $SetAR = \emptyset$
2. calculateCardinal(PRR,D)
3. calculateCardinal(CRR,D)
4. **if** | PRR | = 1 **do begin**
5. **forall** (I_k in I) **do begin**
6. **if** (I_k) $\notin$ PRR and (I_k) $\notin$ CRR **do begin**

STEP 1

7. $AR_i = I_k \rightarrow CRR$

STEP 2

8. calculateCardinal(I_k ,D)
9. $M(AR_i, support)$ = calculateSupport(AR_i)
10. $M(AR_i, confidence)$ = calculateConfidence(AR_i)
11. $M(AR_i, lift)$ = calculateLift(AR_i)

STEP 3

12. **if** $support(M(AR_i)) \geq minSupp$ and $confidence(M(AR_i)) \geq minConf$ **then**
13. $SetAR = SetAR \cup (AR_i, M(AR_i))$
14. **endfor**
15. **if** $|CPR| = 1$ **do begin**
16. **forall** (I_k in I) **do begin**
17. **if** (I_k) $\notin$ PRR and (I_k) $\notin$ CRR **do begin**

STEP 1

18. $AR_i = PRR \rightarrow I_k$

STEP 2

19. calculateCardinal(I_k ,D)
20. $M(AR_i, support)$ = calculateSupport(AR_i)
21. $M(AR_i, confidence)$ = calculateConfidence(AR_i)
22. $M(AR_i, lift)$ = calculateLift(AR_i)

STEP 3

23. **if** $support(M(AR_i)) \geq minSupp$ and $confidence(M(AR_i)) \geq minConf$ **then**
24. $SetAR = SetAR \cup (AR_i, M(AR_i))$
25. **endfor**

STEP 4

26. $SetAR$ = Ranking($SetAR$,RM)
27. **return** $SetAR$

TABLE 5.3: The modified local specialisation anticipation function algorithm.

Input: Database D
Antecedent of the Reference Rule PRR
Consequence of the Reference Rule CRR
Minimum support Thresholds $minSupp$ and $minConf$
Minimum confidence Thresholds $minConf$
Ranking interestingness measure RM
Output: A set of couple (AR, M(AR)) when AR is
an association rule and M(AR) its interestingness measures $SetAR$

1. $SetAR = \emptyset$
2. calculateCardinal(PRR,D)
3. calculateCardinal(CRR,D)
4. **forall** ($I_k \in PRR$) **do begin**

STEP 1
5. $AR_i = PRR\ \ I_k \rightarrow CRR$

STEP 2
6. calculateCardinal($PRR\ \ I_k$,D)
7. $M(AR_i, support)$ = calculateSupport(AR_i)
8. $M(AR_i, confidence)$ = calculateConfidence(AR_i)
9. $M(AR_i, lift)$ = calculateLift(AR_i)

STEP 3
10. **if** $support(M(AR_i)) \geq minSupp$ and $confidence(M(AR_i)) \geq minConf$ **then**
11. $SetAR = SetAR \cup (AR_i$,$M(AR_i))$
12. **endfor**

13. **forall** ($I_k \in CRR$) **do begin**

STEP 1
14. $AR_i = PRR \rightarrow CRR\ \ I_k$

STEP 2
15. calculateCardinal($CRR\ \ I_k$,D)
16. $M(AR_i, support)$ = calculateSupport(AR_i)
17. $M(AR_i, confidence)$ = calculateConfidence(AR_i)
18. $M(AR_i, lift)$ = calculateLift(AR_i)

STEP 3
19. **if** $support(M(AR_i)) \geq minSupp$ and $confidence(M(AR_i)) \geq minConf$ **then**
20. $SetAR = SetAR \cup (AR_i$,$M(AR_i))$
21. **endfor**

STEP 4
22. $SetAR$ = Ranking($SetAR$,RM)

23. **return** $SetAR$

TABLE 5.4: The local generalisation anticipation function algorithm.

5.3.2 Tool Architecture

The IUCEARVis tool proposes to implement the IUCEAR methodology introduced in Chapter 4 Section 4.3. For this purpose, we elaborated a modular and evolving architecture that we designed as shown in Figure 5.1.

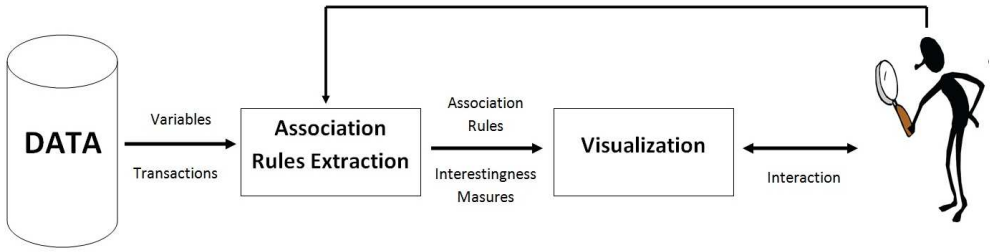


FIGURE 5.1: General architecture of the IUCEARVis tool.

The tool is organised in 3 important parts:

- the data server is the *PostgreSQL* management system of relational databases;
- the iterative process of local association rules generation;
- visualisation and interaction (*OpenGL*).

The process of local association rule generation supports rule extraction. It implements a constraint-based algorithm adapted to the different functions of the module (anticipation functions, filtering functions, etc.). This module is interactive and produces subsets of rules on user request. No rules are produced in advance.

Visualisation and interaction is an interactive process that generates the representation as the user navigates. Scenes generation does not require database access and therefore is low in time consumption. Interactivity is the heart of the IUCEAR methodology. In the association rules field, it is uncommon to find interestingness knowledge in only one search. Thus, an important contribution of our work is to propose an interactive approach. The objective of the interactive process (Figure 5.2) is to help the user find interesting association rules by taking into account user actions. The interactivity also allows the user to come back to his/her actions to finally find interesting knowledge.

Firstly, the interactivity of our approach comes from the process itself that is designed to leave the user the liberty to choose the best action to undertake. Then,

interaction with the different items/association rules representations increases the degree of user interactivity, giving him/her more flexibility to navigate into the representation. Several steps are suggested as follows:

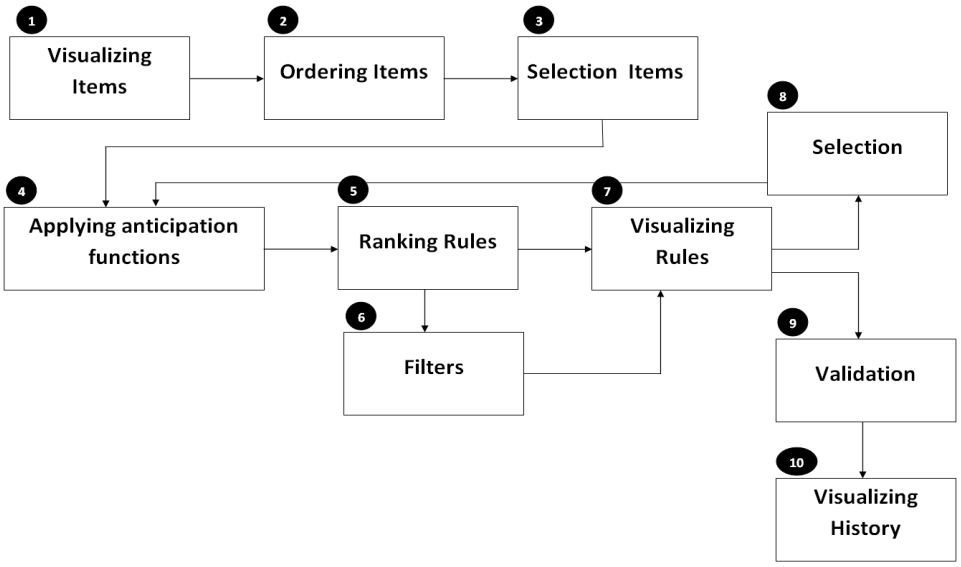


FIGURE 5.2: Interactive process description of IUCEARVis.

1. *Visualising Items*: this first step consists in the development of user expectations. The user choose which items can be interesting and then, should be in an interesting association rule. All the database items are available to the user by means of a pick-list.

2. *Ordering Items*: the ordering operators help the user to choose items. The use of operators increases the level of the tool interactivity. It is the user who chooses which operator he/she wants to apply.

3. *Selection Items*: in this step, the user chooses which items belong to the antecedent and which items belong to the consequent.

4. *Applying anticipation functions*: the second step consists of using the items selected by the user to propose a new association rule set which can be interesting to the user.

5. *Ranking Rules*: many ranking operators can be used to rank the association rules proposed by the anticipation functions. It is the user who chooses which operator to be applied over the association rule set. Letting the user choose which operators will be applied represents an important point in user interactivity. Choosing the operator is choosing what association rules set is visualised first.

6. *Filters*: this phase proposes two filters to be applied over the ranked association rules. These filters can be applied whenever the user needs them.

7. *Visualising Rules*: visualisation is the most important step. It is through visualisation that the user can see the result of his/her actions. An important set of interestingness measures can be visualised to help the user chooses which is the most appropriate decision to take.

8. *Selection*: in this final step, the user chooses a new association rule among the visualised rules set. Then, he/she returns in step 4 in order to apply an anticipation function over the new selected association rule.

9. *Validation*: the user evaluates subjectively the set of rules. In this step, the user can validate an association rule (add to history) or revise his/her information (item selection) and restart the research.

10. *Visualising History*: this step proposes to the user to visualise rules that he/she judges interesting. He/she can delete the rules whenever he/she wants.

5.4 Interaction in IUCEARVis

As we have seen in Chapter 2, VR applications use interaction devices, metaphors and sensory interfaces, to offer the user the ability to interact with virtual entities presented in a virtual context. These interactions are based on usage scenarios (software layer) and interfaces. They allow navigation, selection and manipulation of virtual objects, and application control. Virtual reality techniques are relatively little used in the field of VDM and in particular for association rules visualisation (see Chapter 3). We propose two VR interaction techniques. The first involves bimanual interaction (Figure 5.3) and the second evolves single-handed interaction. In the bimanual interaction, the user uses his dominant hand to select objects in the scene. Both hands are used for objects manipulation (rotation) and translation of the virtual camera (zoom-in and zoom-out). The presence of the user's hands is detected by a motion capture system based on infrared cameras. Reflective stickers are attached to the user's hand.

These two techniques have been implemented at the (1/2, 1, 0) Point of the AIP cube (Chapter 2, Section 2.2): solutions (association rules) are generated semi-automatically through local association rule generating algorithms; solutions are presented in a virtual context, and objects have no autonomy (they are manipulated by the user).

5.4.1 Object Selection and Manipulation

The user uses his/her dominant hand for selection, the non-dominant hand is used to activate selection. When the two user's hands are detected, movement of both hands can change the virtual camera coordinates for rotation (rotate the object relative to its pivot point) and so visualise the object from different points of view. When the

FIGURE 5.3: Bimanual interaction.

user moves his/her hands, he/she makes rotation of the virtual camera: (left and right). When the user moves away his/her hands from each other, he/she performs right rotation. Otherwise, it performs left rotation. In the second interaction technique, the user operates an interaction device: Wiimote^{TM} or a classic mouse. He/she performs the same tasks described previously (selection, rotation) and translation of the virtual camera (zoom-in and zoom-out). The only difference is that the user need to click on a button of the mousse or the Wiimote^{TM} to select an object or to activate the camera movement.

Figure 5.4 illustrates the different possibilities of user actions on the camera. These actions are:

- increase or decrease of the distance between the camera and the object;
- specify the angle of view of the virtual camera.

The automaton of the camera movement (Figure 5.5) is composed of the following states:

- **R** : the camera moves away from the object;
- **I** : the camera returns to the initial configuration;
- **C** : the camera approaches the object.

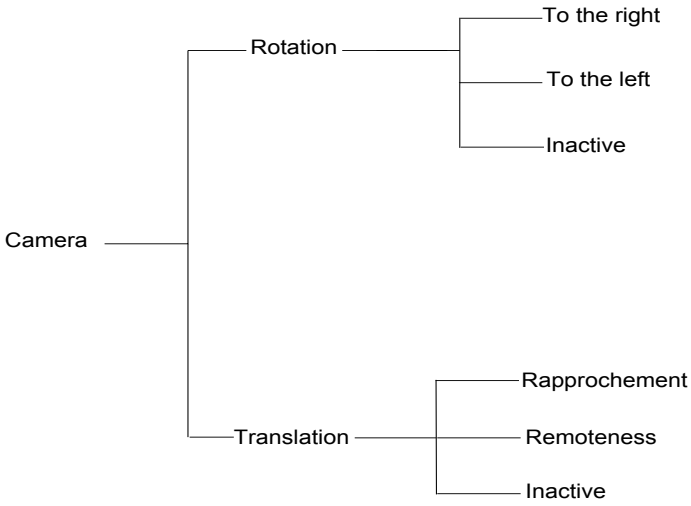


FIGURE 5.4: Illustration of the different possibilities of camera controlled movements.

State changes are triggered via button (mouse button or wiimote^{TM} button) and modelled by the following logical variables:

- T_R : triggering the remoteness of the camera;
- T_C : triggering the approach of the camera.

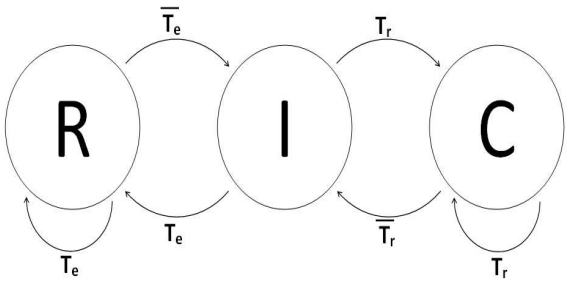


FIGURE 5.5: Automation governing the distance camera - object.

The automation of the camera rotation (Figure 5.6) is composed of the following states :

- **L** : the camera moves in the counterclockwise direction (left);
- **U** : the camera moves up;

- **I** : the camera returns to the initial configuration;
- **R** : the camera moves in the clockwise direction (right);
- **D** : the camera moves down.

The camera is always oriented towards the object. The camera movement is proportional to the offset(ΔX or ΔY) between the position of the mouse at a given moment t and the position of the mouse at the instant of activation of the button of rotation in the one-hand configuration or the distance between the two user hands at a given moment t and the distance between the two user hands at the moment of detection of the second hand in the bimanual configuration. State changes, related to the camera's lateral movement, are triggered by activating the mouse (or Wiimote^{TM}) left button and its movement or by the detection of the two user hands.

Logical variables used for modelling are:

- **ML** : the offset of the cursor to the left;
- **MR** : the offset of the cursor to the right;
- **MU** : the offset of the cursor to up;
- **MD** : the offset of the cursor to down;
- **MB** : activation of rotation (mousse/Wiimote^{TM} left button or the modification of the distance between the user's two hands).

At the deactivation of the rotation (deactivation of the mouse/Wiimote^{TM} left button or the detection of only one user hand), the current state becomes the camera's initial state.

5.4.2 System Control

IUCEARVis proposes many interactive operations, such as changing a constraint, moving from one interface to an other,etc. It appears that the use of a system control is increasingly important as the complexity of the application increases– in terms of the number of operations. While it is entirely possible to be satisfied with a rudimentary or nonexistent interface when the only possible action is to move and rotate a virtual object, it is quite different when the interaction possibilities increase. On the one hand, it is necessary to allow the user to activate them, but, in the other hand, it is also preferable that it can be done easily and efficiently. The role of system control is to offer opportunities, but also and above is to provide efficiency and sensibility. For this purpose, we propose to represent all the controls as buttons. This solution is intuitive and easy to use; the user just has to click on a button to activate a control.

The schema relative to the activation of different features of interaction with the extraction association rule algorithm is shown in Figure 5.7. To facilitate schema understanding we have added a different interfaces with which the user interacts. The schema is composed of the following states:

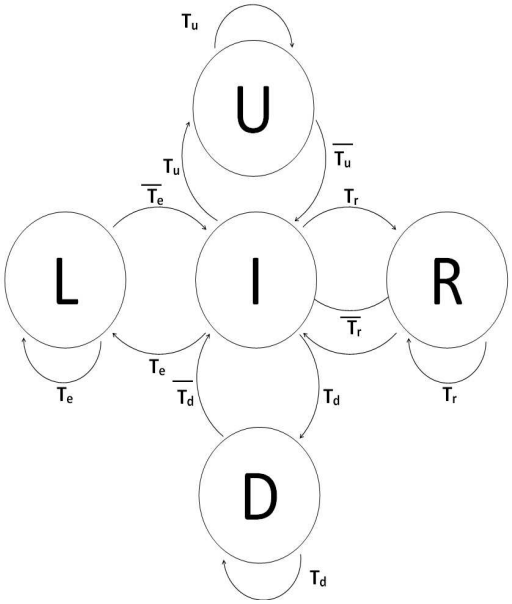


FIGURE 5.6: Automaton governing camera rotation.

- Interestingness measure calculation;
- **Item selection interface**;
- Association rule extraction;
- **Association rule exploration interface**;
- Interesting Association rules list;
- **History visualisation interface**.

As we saw earlier, the state changes are triggered by clicking on a button. These buttons are modelled by the following variables state :

- Add an item to the rule;
- Delete an item from the rule;
- Change the support threshold;
- Change the confidence threshold;
- Specify whether the item will be added to the antecedent or consequent;
- Set a new reference rule;
- Display the item selection interface;
- Display the history visualisation interface;

- Modification the reference rule;
- Add an association rule to the history list;
- Modify the association rule exploration interface.

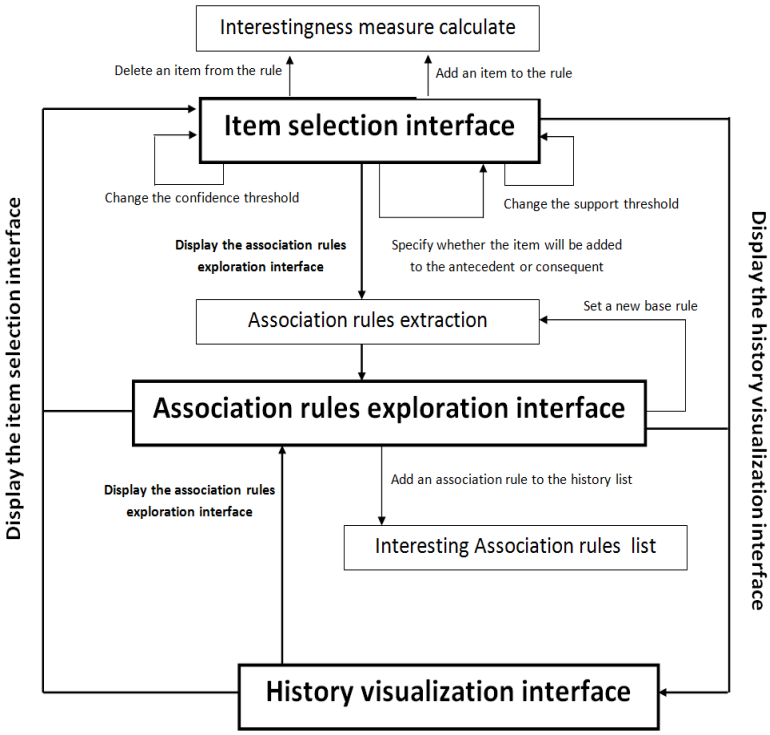


FIGURE 5.7: Illustration of the interaction possibilities with the extraction algorithms.

5.5 Case Study

This study is based on psychological profile database, provided by *PerformanSe SA*. This company designs software for human resource management decision support dedicated to behavioural evaluation and motivation in the workplace. The profile database contains the profiles of 4065 persons (reference population of French adults). The profiles are described by ten behavioural traits (Table 5.5). Each behavioural trait is encoded by a qualitative variable with three modalities coded as {+, 0, -}: strongly affirmed, moderately affirmed, and weakly affirmed.

We are interested in rigorous people. We begin by studying behavioural traits involved by searching for independence (AFL = -) and strictness (CON = +). Firstly, we discovered that these criteria are not very common in the data by means of the item selection interface (Figure 5.8). Then, we realised that it is a low quality rule

Behavioral trait	**Representation**
Extroversion	$E \in \{-, 0, +\}$
Pugnacity	$P \in \{-, 0, +\}$
Nevrosism	$N \in \{-, 0, +\}$
ACHievement	$ACH \in \{-, 0, +\}$
CLeVerness	$CLV \in \{-, 0, +\}$
CONscienciousness	$CON \in \{-, 0, +\}$
Emotional STability	$EST \in \{-, 0, +\}$
LEaDership	$LED \in \{-, 0, +\}$
AFfiLiation	$AFL \in \{-, 0, +\}$
RECeptivity	$REC \in \{-, 0, +\}$

TABLE 5.5: Behavioral traits

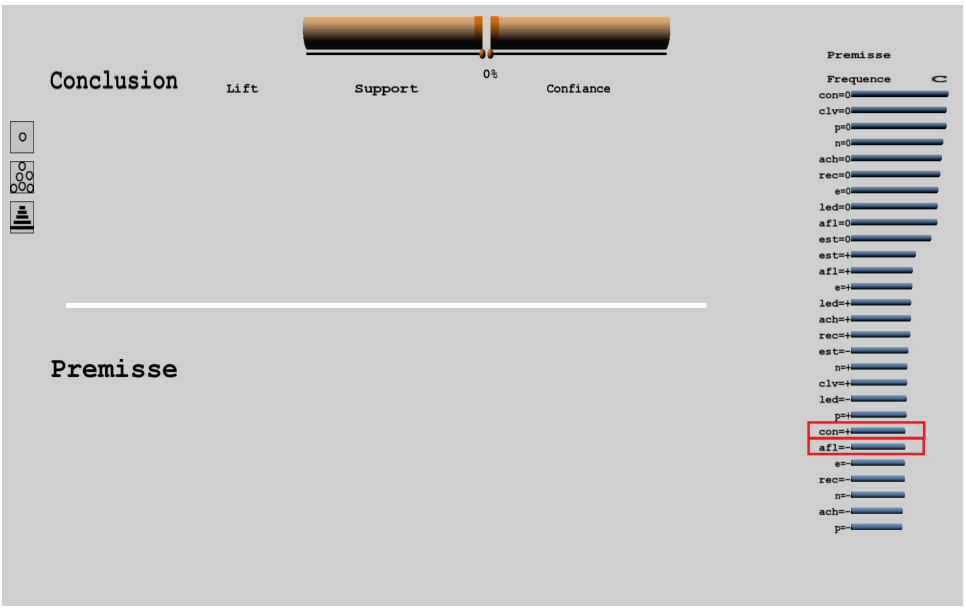


FIGURE 5.8: Illustration 1.

(support = 9%, confidence = 34%, and lift = 1.29) (Figure 5.9). We explored the rules proposed by the anticipation functions by means of the association rule exploration interface. We looked for rules that have the greatest confidence. So, we chose to order the proposed association rules by confidence. There are four rules with high confidence (> 75%). The negative Information Gain (GI) of the item (AFL = -) draws our attention (Figure 5.10): the search for independence (AFL = -) is a bad criterion to explain strictness (CON = +). We eliminate (AFL = -) from the

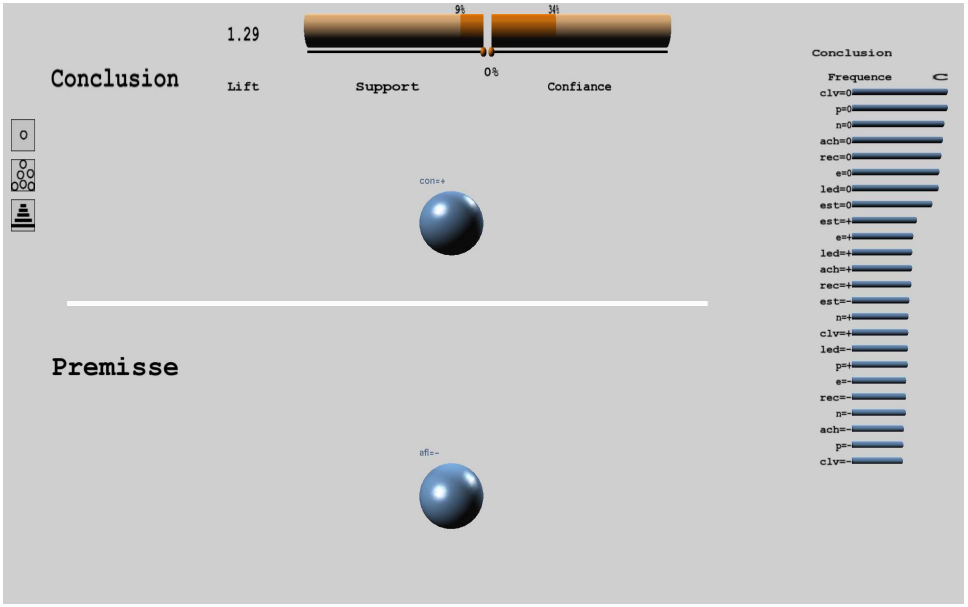


FIGURE 5.9: Illustration 2.

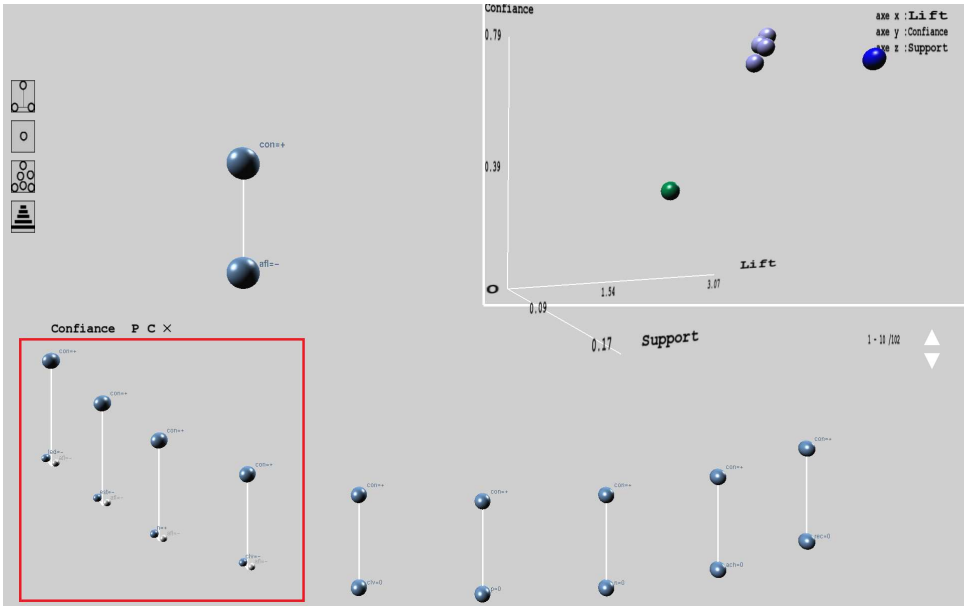


FIGURE 5.10: Illustration 3.

reference rule and choose to add (LED = -): motivation for protection. The system proposes 102 rules. The 10 first rules are all good rules (72% ≤confidence ≤ 95%, 2% ≤support ≤ 12% and lift ≥ 2.7) (Figure 5.11).

One of the best rules is: (CLV = - , LED = - → CON = + ;confidence = 92% , support= 12%, lift = 3.35) which means that lack of cleverness and seeking protection are good predictors of strictness (Figure 5.12). This is quite a good rule, but we want

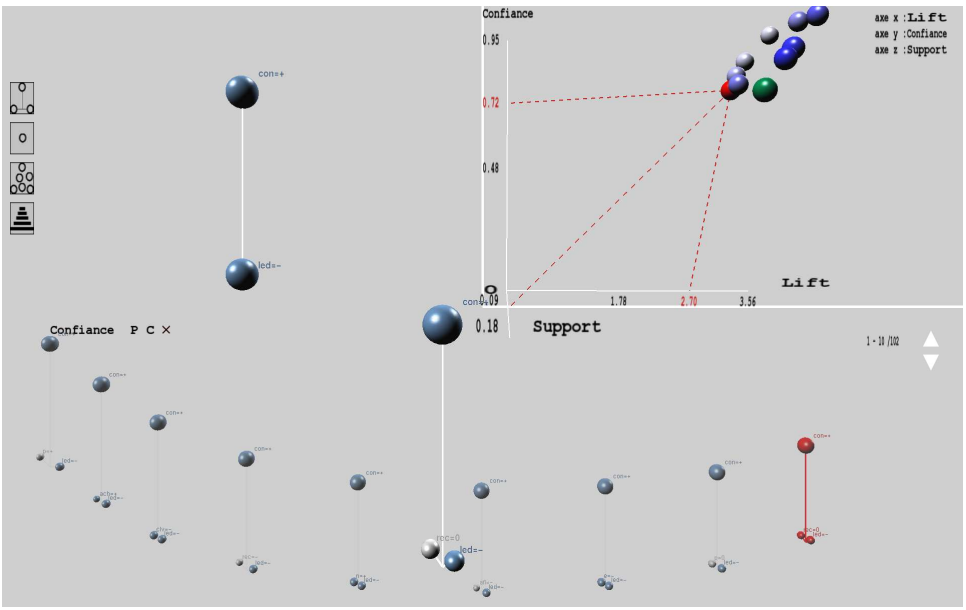


FIGURE 5.11: Illustration 4.

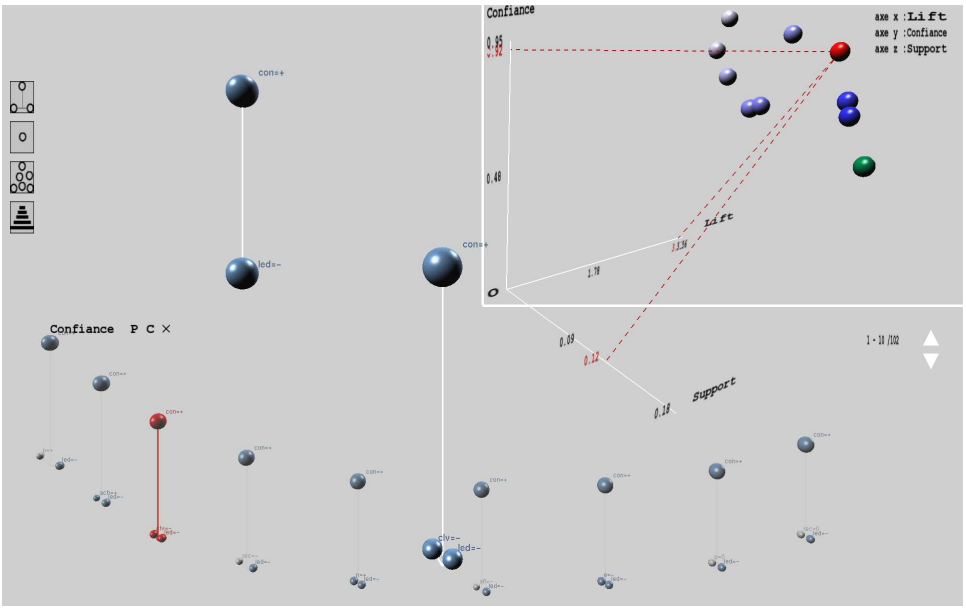


FIGURE 5.12: Illustration 5.

to verify whether other characteristics could better predict strictness. To do so, we select (CLV = - , LED = - → CON = +) as a new reference rule. The new rules set contains 68 rules. Two rules catch our attention: (CLV = - , LED = -, E = - → CON = +; confidence = 96% , support= 10%, lift = 3.60) and (CLV = - , LED = -, ACH = + → CON = +; confidence = 96% , support= 7%, lift = 3.61). It is the only rules that does not contain items with negative *Information Gain* (Figure 5.13).

Achievement and introversion are good predictors for strictness. We add these rules

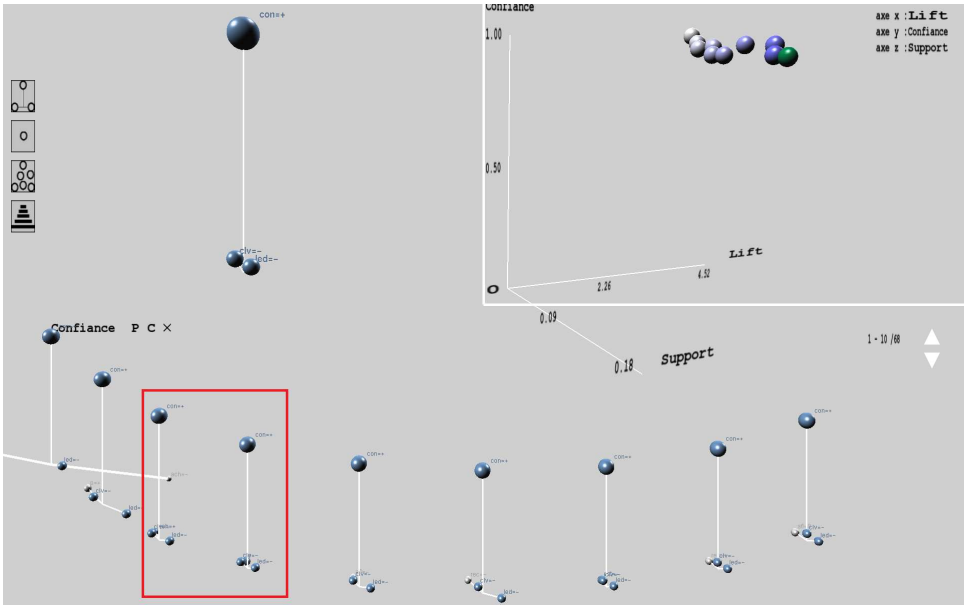


FIGURE 5.13: Illustration 6.

to the list of interesting rules (visualised by the history interface). We select the rule (CLV = - , LED = -, E = - → CON = +) as a new reference rule to verify if other characteristics could predict strictness. We look for rules that maximise the three rule interestingness measures. Thus, we classify rules with the aggregate measure. The metaphor (distance between the antecedent and the consequence) allows us to distinguish the rules with good confidence (Figure 5.14). Several new items appear. They are little correlated with items (CLV = - , LED = -, E = -). These items are (ACH = + , N = +, EST = -). The items (N = +) and (EST = -) appear also in the consequent of two rules that have good interestingness measures: (CLV = - , LED = -, E = - → CON = + , EST = -) and (CLV = - , LED = -, E = -→ CON = + , N = +). We can conclude that lack of cleverness, motivation for protection and introversion can explain both strictness and questioning and strictness and anxiety. To verify other criteria verified by lack of cleverness, motivation for protection and introversion (CLV = - , LED = -, E = -) we applied the filter ***same antecedent*** to the rule (CLV = - , LED = -, E = - → CON = +). No rule has an equal or higher quality than the reference rule (Figure 5.15).

Finally, we found five interesting rules :

1. (CLV = - , LED = -) → (CON = +) (confidence = 92% , support= 12%, lift = 3.35)

2. (CLV = - , LED = -, E = -) → (CON = +) (confidence = 96% , support= 10%, lift = 3.60)

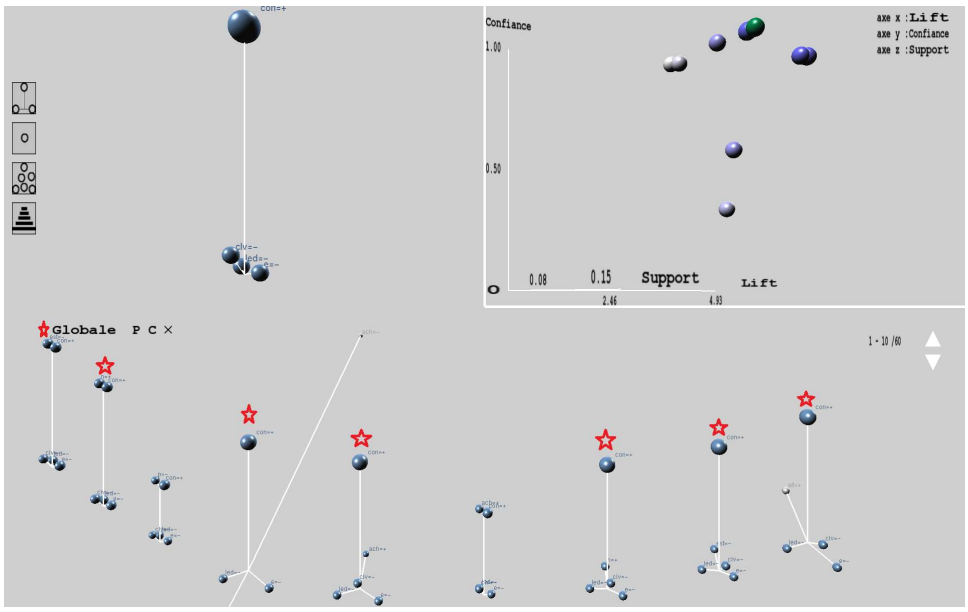


FIGURE 5.14: Illustration 7.

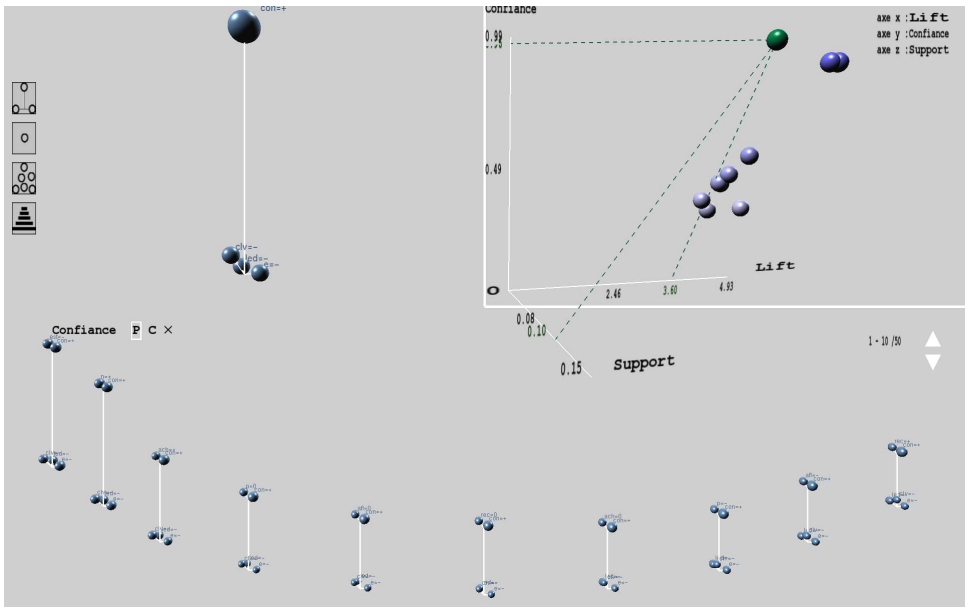


FIGURE 5.15: Illustration 8.

3. (CLV = - , LED = -, ACH = +) → (CON = +) (confidence = 96% , support= 7%, lift = 3.61)

4. (CLV = - , LED = -, E = -) → (CON = + , EST = -)(confidence = 89% , support= 9%, lift = 4.93)

5. (CLV = - , LED = -, E = -) → (CON = + , N = +)(confidence = 89% ,

support= 9%, lift = 4.98)

5.6 Conclusion

This Chapter was consecrated to the implementation of the IUCEARVis tool. IUCEARVis allows user (s) to interactively generate, explore, and validate the whole set of association rules by means of an iterative process.

Firstly, we have mainly detailed the algorithms used for rule generation. We adapted the extraction of rules to the rule post-processing interactive character of the IUCEAR methodology. For that purpose, we developed specific algorithms for local association rule extraction. These algorithms generate rules based on a *reference rule* constructed by the user. They exploit constraints engendered by the anticipation functions and are specifically optimised for these constraints. With the syntactic constraints, the space research is drastically reduced; the algorithms are polynomial depending on the number of items. This local approach gives the possibility to overcome the limits of the exhaustive algorithms, for instance the *Apriori* algorithm, mainly the number of generating rules can be very large. In particular, even very specific rules can be extracted with local algorithms.

Then, we detailed how the modules which compose the tool were implemented. We discussed the choices that we made during development, in particular concerning software technology.

Regarding interaction techniques, we considered two approaches for the selection and manipulation of virtual objects. The first technique evolves bimanual interaction and the second evolves single-handed interaction. In the first technique, the presence of the user's hands is detected by a motion capture system based on infrared cameras. In the second one, the user operates an interaction device: Wiimote^{TM} or a classic mouse.

Testing IUCEARVis on real data shows that the tool helps discover interesting rules of good quality, and in particular the locally dominant rules (the best rules in the explored region).

Conclusion and Perspectives

This thesis is concerned with the merging of two active research domains: Knowledge Discovery in Databases (KDD), and more precisely the Association Rule Mining and Virtual Reality techniques.

Data mining algorithms generate association rules in large quantities so that the user can not generally use them directly. To identify interesting useful knowledge, it is necessary in the output of the data mining algorithms to carry out a post-processing of association rules, consisting of a second search operation. Although data mining is done automatically by combinatorial algorithms, finding interesting rules is done by the user. This is a tedious task in practice.

Contributions

The contributions of this thesis can be summarised as follows:

- we propose a new classification for VDM techniques based on both 3D representations and interaction techniques;
- we propose a new metaphor for association rule representation;
- we establish a methodology for interactive visualisation of association rules;
- we develop specific algorithms for local extraction of association rules;
- we create a tool for the interactive visualisation of association rules in virtual environments.

Classification for VDM techniques based on both 3D representations and interaction techniques

We present a new classification of VDM tools based on 3 dimensions: visual representations, interaction techniques, and KDD tasks. The proposed taxonomy takes into account both visual representation and interaction techniques in the context of data mining applications. Existing metaphors for visualisation and interaction can be classified under the new system, enabling designers to more easily compare metaphors to see exactly how they are different and similar to each other.

This classification looks at some representative tools for performing different KDD tasks, e.g., pre-processing and post-processing (classification, clustering and association rules). Different tables summarise the main characteristics of the reported VDM tools with regard to visual representations and interaction techniques. Other relevant information such as interaction actions (navigation, selection and manipulation, and system control), input-output devices (CAVEs, mice, hand trackers, etc.) presentation (3D representation or VR representation) and year of creation is also reported.

New metaphor for association rule representation

We propose a new visualisation metaphor for association rules. This new metaphor represents attributes which make up the antecedent and the consequent, the contribution of each one to the rule, and the correlations between each pair of the antecedent and each pair of the consequent.

In this context, we developed 3D visual representations to overcome some limitations of 2D representations. This new metaphor is based on the principles of information visualisation for effective visual representation [24]. The association rule has a molecular representation based on a Node-Link Algorithm which is used to calculate item positions in the 3D space. A validation study was carried out for the evaluation of four different representations of association rules with different interaction techniques and visualisation interfaces. Although this test requires deepening further work, we have shown that this metaphor allows the discovery of interesting relationships among items that are not visible with the other metaphors.

Methodology for interactive visualisation of association rules: IUCARE

We developed an interactive visualisation methodology for association rules, named Interactive User-Centred Association Rules Exploration (IUCARE). It is designed to assist the user when facing large sets of rules taking into account his/her capacity to process information. This user-centred methodology can really assist the user when searching for interesting knowledge in a rules set by combining the three main approaches that are traditionally offered to facilitate the rules post-processing: rules interestingness measure, interactivity, and visualisation. The user selects interesting knowledge and the system proposes sets of rules based on this knowledge. Then, the user can navigate in this rules set using interactive visualisation of rules and their interestingness measures. Thus, the user directs a series of local visual explorations based on his/her interest for the rules. The user does not deal with a large set of rules, but he/she explores it subset by subset. This approach is based on the cognitive principles of Montgomery [200]. Based on this principle, we developed the *anticipation functions*. These functions allow for the extraction of small subsets of rules based on items selected by the user. The items selected by the user can be changed at any time during the navigation. These functions provide genuine originality to our

methodology. Another original facet of the IUCARE methodology, is the possibility to note the interesting association rules and visualise them on a dedicated interface. This interface gives the user the ability to quickly find, view, and compare the rules that he/she considered interesting during his/her previous explorations.

Algorithms for local association rules extraction

We propose adopting the rule extraction from the interactive nature of rules post-processing. For this purpose we developed specific algorithms for local association rules extraction. They are constraints-based algorithms that extract only the rule sets that the user wishes to view in his/her exploration. Thus, the user operates role of a heuristically integrated within the process of association rules extraction. The use of powerful constraints provides the extraction of very few rules compared to the exhaustive algorithms. In addition, even the very specific rules (regarding a very small portion of the data) can be extracted with the local extraction algorithms.

The interactive visualisation tool IUCAREVis

We develop an interactive tool for association rules visualisation in virtual reality: IUCAREVis. This tool implements the three approaches described above: new metaphor for association rule representation, IUCARE methodology, and algorithms for local association rules extraction. IUCAREVis is based on a virtual reality visualisation and intuitive interaction. This representation enhances rules interestingness measures and therefore facilitates the recognition of the most suitable rules. IUCAREVis incorporates two interaction techniques: bi-manual techniques and a one-hand interaction technique. These interactions allow the user to navigate among the sets of rules.

Perspectives

Development of constraint-based algorithms with memory

A higher performance strategy for local rules extraction can be achieved. Saving intermediate results will avoid the generation of the same association rule several times. Extraction with memory is an intermediate solution between exhaustive extraction (full memory) and constraint-based extraction (without memory). This solution will decrease the algorithms execution time.

IUCAREVis validation

The tool can be tested by an expert with the same or other data. This would confirm

our hypothesis with an expert in rule validation. It will allows us to envisage new interactions, anticipation functions, filters, etc, based on expert advice or by analysing the exploration history of IUCAREVis. Then, it is possible to validate the interaction and visualisation interfaces proposed in IUCAREVis as we did for the validation of association rule metaphors. An experimental protocol may consist of asking several user groups using different configurations of IUCAREVis (monoscopic/stereoscopic visualisation, tracking / Wiimote interaction devices, etc) to perform the same exploration tasks on the same data. This experiment can be used also to compare IUCAREVis tool to other association rules exploration tools. Such experiment would allow us to compare the accuracy and the speed of the user responses.

The use of ontologies

Rules exploration can be improved by incorporating the possibility of using one or more item hierarchies. This allows the user to specialise or to generalise the rules not only by adding or removing items, but also by going down or up the items in the hierarchy. This approach could be generalised by allowing the introduction of external knowledge, such as ontology of the subject field, information to explain the data, or even user annotation. Such associations between knowledge extracted from the data and external knowledge is ambitious, but we think it would facilitate the appropriation of knowledge by the user.

A 3D marking menu

The application control can be facilitate with the use of a 3D marking menu. On the one hand, the use of a menu displayed on the user request allows to free the view; all the commands will be eliminated from the displayed interface and will be displayed only if the user needs to use them. On the other hand, a 3D marking menu can be more ergonomic, more aesthetic, and easier to use. A marking menu can be used without a visual feedback of the graphical menu (user memorise the gestures necessary for the activation of a given command), than the user is able to activate a command without taking his eyes or attention from the main task. The menu can be manipulated by the user non-dominant hand, thus remains the dominant hand free for the main tasks of objects selection and manipulation.

A collaborative work system

The association rules mining is preferably performed by different background experts such as databases, marketing, statistics, machine learning, etc. Very few people are expert in all these domains. So it is not uncommon that multiple expert should work together on the same mining process. To work together all the experts should

be present in the same location. The development of a VE for collaborative association rules mining in which each expert is represented by an avatar allows experts to collaborate without necessarily be present in the same place. Each expert should be able to identify the other experts actions.

References

[1] *www.palantirtech.com.*

[2] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207 – 216, 1993.

[3] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, 1994.

[4] Adel Ahmed, Tim Dwyer, Michael Forster, Xiaoyan Fu, Joshua Ho, Seok-Hee Hong, Dirk Koschtzki, Colin Murray, Nikola S. Nikolov, Ronnie Taiband Alexandre Tarassov, and Kai Xu. Geomi: Geometry for maximum insight. *Graph Drawing*, 3843:468–479, 2006.

[5] Adel Ahmed and Peter Eades. Automatic camera path generation for graph navigation in 3d. In *Proceedings of the 2005 Asia-Pacific symposium on Information visualisation (APVis '05)*, pages 27–32. Australian Computer Society, Inc., 2005.

[6] Y. Aitsiselmi and N. S. Holliman. Using mental rotation to evaluate the benefits of stereoscopic displays. *Stereoscopic Displays and Applications*, 7237, 2009.

[7] Robert Amar, James Eagan, and John Stasko. Low-level components of analytic activity in information visualization. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, pages 15–, Washington, DC, USA, 2005. IEEE Computer Society.

[8] Ayman Ammoura, Osmar R. Zaïane, and Yuan Ji. Immersed visual data mining: Walking the walk. In *Proceedings of the 18th British National Conference on Databases (BNCOD'01)*, pages 202–218. Springer-Verlag, 2001.

[9] Carlos Andujar, Marta Fairen, and Ferran Argelaguet. A cost-effective approach for developing application-control guis for virtual environments. In *Proceedings of the 3D User Interfaces*, 3DUI '06, pages 45–52, Washington, DC, USA, 2006. IEEE Computer Society.

[10] Mihael Ankerst. *Visual Data Mining.* PhD thesis, Institute for Computer Science Database and Information Systems, University of Munich, 2001.

[11] Laura Lynn Arns and Carolina c. A new taxonomy for locomotion in virtual environments. In *Proceedings second Virtual Reality International Conference*, VRIC'02. Iowa State University, 2000.

[12] Mafruz Zaman Ashrafi, David Taniar, and Kate Smith. Redundant association rules reduction techniques. *Advances in Artificial Intelligence*, pages 254–263, 2005.

[13] Daniel Asimov. The grand tour: a tool for viewing multidimensional data. *SIAM journal on scientific and statistical computing*, 6(1):128–143, 1985.

[14] H. Azzag, F. Picarougne, C. Guinot, and G. Venturini. Vrminer: a tool for multimedia databases mining with virtual reality. In J. Darmont and O. Boussaid, editors, *Processing and Managing Complex Data for Decision Support*, pages 318–339, 2005.

[15] B. Baesens, S. Viaene, and J. Vanthienen. Post-processing of association rules. In *Workshop on Post-Processing in Machine Learning and Data Mining: Interpretation, visualization, integration, and related topics with in Sixth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 20–23, 2000.

[16] J.P. Barthlemy. A model of selection by aspects. *Acta Psychologica*, 79(1):119, 1992.

[17] Sebastian Baumgärtner, Achim Ebert, Matthias Deller, and Stefan Agne. 2d meets 3d: a human-centered interface for visual data exploration. In *Extended abstracts on Human factors in computing systems*, CHI '07, pages 2273–2278. ACM, 2007.

[18] Jr. Bayardo and Roberto J. Efficiently mining long patterns from databases. *ACM SIGMOD Record*, 27:85–93, June 1998.

[19] Roberto J. Bayardo, Jr. and Rakesh Agrawal. Mining the most interesting rules. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '99, pages 145–154. ACM, 1999.

[20] Russell Beale. Supporting serendipity: Using ambient intelligence to augment user exploration for data mining and web browsing. *International Journal of Human-Computer Studies*, 65(5):421–433, 2007.

[21] B.G. Becker. Volume rendering for relational data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '97)*, pages 87–91, 1997.

[22] Steffi Beckhaus, Kristopher J. Blom, and Matthias Haringer. Intuitive, hands-free travel interfaces for virtual environments. In *IEEE Workshop in new directions in 3D user interfaces*, pages 57–60, 2005.

[23] C. Beilken and M Spenke. Interactive data mining with infozoom –the medical data set. In *Workshop Notes on Discovery Challenge*, PKDD, pages 49–54, 1999.

[24] Jacques Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, 1967.

[25] Inderpal Bhandari. Attribute focusing: machine-assisted knowledge discovery applied to software production process control. *Knowledge Acquisition*, 6(3):271294, 1994.

[26] Mark Billinghurst, Sisinio Baldis, Lydia Matheson, and Mark Philips. 3d palette: a virtual reality content creation tool. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 155–156, 1997.

[27] Mark Billinghurst, Hirkazu Kato, and Ivan Poupyrev. The magicbook&#151moving seamlessly between reality and virtuality. *IEEE Computer Graphics and Applications*, 21(3):6–8, May 2001.

[28] Julien Blanchard, Fabrice Guillet, and Henri Briand. Exploratory visualization for association rule rummaging. In *Workshop on Multimedia Data Mining*, MDM-03, pages 107–114, 2003.

[29] Julien Blanchard, Fabrice Guillet, and Henri Briand. Interactive visual exploration of association rules with rule-focusing methodology. *Knowledge and Information Systems*, 13(1):43–75, 2007.

[30] Julien Blanchard, Fabrice Guillet, and Pascale Kuntz. *Post-Mining of Association Rules: Techniques for Effective Knowledge Extraction*, chapter Semantics-Based Classification of Rule Interestingness Measures, pages 56–79. 2009.

[31] Julien Blanchard, Bruno Pinaud, Pascale Kuntz, and Fabrice Guillet. A 2d-3d visualization support for human-centered rule mining. In *Computers & Graphics*, 2007.

[32] Richard A. Bolt. Put-that-there: Voice and gesture at the graphics interface. *SIGGRAPH Computer Graphics*, 14:262–270, 1980.

[33] Francesco Bonchi, Fosca Giannotti, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Roberto Trasarti. On interactive pattern mining from relational databases. In *Proceedings of the 5th international conference on Knowledge discovery in inductive databases*, KDID'06, pages 42–62. Springer-Verlag, 2007.

[34] Francesco Bonchi, Fosca Giannotti, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Roberto Trasarti. A constraint-based querying system for exploratory pattern discovery. *Information Systems*, 34:3–27, March 2009.

[35] Laroussi Bouguila, Florian Evequoz, Michele Courant, and Beat Hirsbrunner. Walking-pad: a step-in-place locomotion interface for virtual environments. In *Proceedings of the 6th international conference on Multimodal interfaces*, ICMI '04, pages 77–81, New York, NY, USA, 2004. ACM.

[36] D. A. Bowman, C. A. Wingrave, J. M. Campbell, V. Q. Ly, and C. J. Rhoton. Novel uses of pinch gloves for virtual environment interaction techniques. *Virtual Reality*, 6(3):122–129, 2001.

[37] Doug A. Bowman, Jian Chen, Chadwick A. Wingrave, John Lucas, Andrew Ray, Nicholas F. Polys, Qing Li, Yonca Haciahmetoglu, Ji-Sun Kim, Seonho Kim, Robert Boehringer, and Tao Ni. New directions in 3d user interfaces. *The International Journal of Virtual Reality*, 6(2):3–14, 2006.

[38] Doug A. Bowman and Bernd Frohlich. New directions in 3d user interfaces. In *Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality*, VR '05, pages 312–, Washington, DC, USA, 2005. IEEE Computer Society.

[39] Doug A. Bowman and Larry F. Hodges. Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments. *Journal of Visual Languages & Computing*, 10(1):3753, 1999.

[40] Doug A. Bowman, David Koller, and Larry F. Hodges. Travel in immersive virtual environments: An evaluation of viewpoint motion control techniques. In *Proceedings of the Virtual Reality Annual International Symposium*, pages 45–52, 1997.

[41] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. An introduction to 3-d user interface design. *Presence: Teleoper. Virtual Environ.*, 10(1):96–108, 2001.

[42] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.

[43] Doug A. Bowman and Chadwick A. Wingrave. Design and evaluation of menu systems for immersive virtual environments. In *Proceedings of the Virtual Reality 2001 Conference*, VR '01, pages 149–, Washington, DC, USA, 2001. IEEE Computer Society.

[44] Douglas A. Bowman. *Interaction Techniques For Common Tasks In Immersive Virtual Environments - Design, Evaluation, And Application*. PhD thesis, Georgia Institute of Technology, 1998.

[45] Ronald J. Brachman and Tej Anand. Advances in knowledge discovery and data mining. chapter The process of knowledge discovery in databases, pages 37–57. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.

[46] Daniele Braga, Alessandro Campi, Mika Klemettinen, and Pier Luca Lanzi. Mining association rules from xml data. In *Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery*, DaWaK 2000, pages 21–30, 2002.

[47] Richard Brath, Mike Peters, and Robin Senior. Visualization for communication: The importance of aesthetic sizzle. In *Proceedings of the Ninth International Conference on Information Visualisation*, pages 724–729. IEEE Computer Society, 2005.

[48] Sergey Brin, Rajeev Motwani, and Craig Silverstein. Beyond market baskets: generalizing association rules to correlations. *ACM SIGMOD Record*, 26:265–276, June 1997.

[49] Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, and Shalom Tsur. Dynamic itemset counting and implication rules for market basket data. *ACM SIGMOD Record*, 26(2):255–264, 1997.

[50] Dario Bruzzese and Paolo Buono. Combining visual techniques for association rules exploration. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '04, pages 381–384, New York, NY, USA, 2004. ACM.

[51] Dario Bruzzese and Cristina Davino. Visual mining of association rules. In Springer Berlin / Heidelberg, editor, *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*, volume 4404, pages 103–122. S.J. Simoff et al (Eds), 2008.

[52] Linas Bukauskas and Michael Bhlen. Observer relative data extraction. In *In Proceedings of the International Workshop on Visual Data Mining, in conjunction with ECML/PKDD2001, 2nd European Conference on Machine Learning and 5th European http://www.opensg.org/ Conference on Principles and Practice of Knowledge Discovery in Data*, pages 1–2, 2001.

[53] Cody Buntain. 3d ontology visualization in semantic search. In *Proceedings of the 46th Annual Southeast Regional Conference*, pages 204–208, 2008.

[54] Grigore Burdea and Philippe Coiffet. *La ralit virtuelle*. 1993.

[55] Doug Burdick. Mafia: A maximal frequent itemset algorithm for transactional databases. In *Proceedings of the 17th International Conference on Data Engineering*, pages 443–, Washington, DC, USA, 2001. IEEE Computer Society.

[56] Jean-Marie Burkhardt. Immersion, ralisme et prsence dans la conception et l'valuation des environnements virtuels. *Psychologie franaise*, 48(2):35–42, 2003.

[57] Jeff Butterworth, Andrew Davidson, Stephen Hench, and Marc. T. Olano. 3dm: a three dimensional modeler using a head-mounted display. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 135–138, 1992.

[58] Jochen c, Ulrich Güntzer, and Gholamreza Nakhaeizadeh. Algorithms for association rule mining: a general survey and comparison. *ACM SIGKDD Explorations Newsletters*, 2:58–64, June 2000.

[59] Yang Cai, Richard Stumpf, Timothy Wynne, Michelle Tomlinson, Daniel Sai Ho Chung, Xavier Boutonnier, Matthias Ihmig, Rafael Franco, and Nathaniel Bauernfeind. Visual transformation for interactive spatiotemporal data mining. *Knowledge and Information Systems*, 13(2):119–142, 2007.

[60] Stuart K. Card, Jock D. Mackinlay, and Ben Schneiderman. *Readings in information visualization : using vision to think*. Morgan Kaufmann publishers, 1999.

[61] Stuart K. Card, George G. Robertson, and William York. The webbook and the web forager: an information workspace for the world-wide web. In *Proceedings of the SIGCHI conference on Human factors in computing systems: common ground*, CHI '96, pages 111–ff., New York, NY, USA, 1996. ACM.

[62] C. Melody Carswell, Sylvia Frankenberger, and Donald Bernhard. Graphing in depth: Perspectives on the use of three-dimensional graphs to represent lower-dimensional data. *Behaviour & Information Technology.*, 10(6):459–474, 1991.

[63] Aaron Ceglar, John F. Roddick, and Paul Calder. Managing data mining technologies in organizations. chapter Guiding knowledge discovery through interactive data mining, pages 45–87. IGI Publishing, Hershey, PA, USA, 2003.

[64] Sharma Chakravarthy and Hongen Zhang. Visualization of association rules over relational dbmss. In *Proceedings of the 2003 ACM symposium on Applied computing*, SAC '03, pages 922–926. ACM, 2003.

[65] D. Chamaret. *Plate-forme de ralit virtuelle pour ltude de laccessibilit et de lextraction de lampes sur prototype virtuel automobile.* PhD thesis, Universit dAngers, 2010.

[66] Sarah S. Chance, Florence Gaunet, Andrew C. Beall, and Jack M. Loomis. Locomotion mode affects the updating of objects encountered during travel: The contribution of vestibular and proprioceptive inputs to path integration. *Presence: Teleoper. Virtual Environ.*, 7(2):168–178, April 1998.

[67] Pritam Chanda, Jianmei Yang, Aidong Zhang, and Murali Ramanathan. On mining statistically significant attribute association information. *SIAM*, pages 141–152, 2010.

[68] Herman Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68(342):361–368, 1973.

[69] W. Cheung and O.R. Zaiane. Incremental mining of frequent patterns without candidate generation or support constraint. In *Proceedings. Seventh International Database Engineering and Applications Symposium*, pages 111–116, 2003.

[70] Ed H. Chi and John T. Riedl. An operator interaction framework for visualization systems. In *Proceedings of IEEE Symposium on Information Visualization*, INFOVIS '98, pages 63–70. IEEE Computer Society, 1998.

[71] E.H. Chi. A taxonomy of visualization techniques using the data statereference model. In *Proceedings of IEEE Symposium on Information Visualization*, InfoVis'00, pages 69–75, 2000.

[72] Maria Cristina Ferreira de Oliveira Claudio Haruo Yamamoto and Solange Oliveira Rezende. *Post-Mining of Association Rules: Techniques for Effective Knowledge Extraction*, chapter Visualization to Assist the Generation

and Exploration of Association Rules, pages 224–245. Information Science Reference, 2009.

[73] W S Cleveland and R McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):531–554, 1984.

[74] William S. Cleveland. *Visualizing Data*. Hobart Press, 1993.

[75] Andy Cockburn and Bruce McKenzie. Evaluating the effectiveness of spatial memory in 2d and 3d physical and virtual environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '02, pages 203–210. ACM, 2002.

[76] F. Coenen, P. Leng, and S. Ahmed. Data structure for association rule mining: T-trees and p-trees. *IEEE Transactions on Knowledge and Data Engineering*, 16(4):774–778, 2004.

[77] K. Coninx, F. Van Reeth, and E. Flerackers. A hybrid 2d / 3d user interface for immersive object modeling. In *Proceedings of the 1997 Conference on Computer Graphics International*, CGI '97, pages 47–, Washington, DC, USA, 1997. IEEE Computer Society.

[78] Brookshire D. Conner, Scott S. Snibbe, Kenneth P. Herndon, Daniel C. Robbins, Robert C. Zeleznik, and Andries van Dam. Three-dimensional widgets. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, I3D '92, pages 183–188, New York, NY, USA, 1992. ACM.

[79] Olivier Couturier, Tarek Hamrouni, Sadok Ben Yahia, and Engelbert Mephu Nguifo. A scalable association rule visualization towards displaying large amounts of knowledge. In *: Proceedings of the 11th International Conference Information Visualization*, pages 657–663, Washington, DC, USA, 2007. IEEE Computer Society.

[80] Olivier Couturier, José Rouillard, and Vincent Chevrin. An interactive approach to display large sets of association rules. In *Proceedings of the 2007 conference on Human interface: Part I*, pages 258–267, Berlin, Heidelberg, 2007. Springer-Verlag.

[81] Chad Creighton and Samir Hanash. Mining gene expression databases for association rules. *Bioinformatics*, 19(1):79–86, 2003.

[82] Carolina Cruz-Neira, Daniel J. Sandin, and Thomas A. DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the cave. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, pages 135–142, 1993.

[83] Lawrence D. Cutler, Bernd Fröhlich, and Pat Hanrahan. Two-handed direct manipulation on the responsive workbench. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, I3D '97, pages 107–114, New York, NY, USA, 1997. ACM.

[84] R. Dachselt and M. Hinz. Three-dimensional widgets revisited: towards future standardization. In *Proceedings of IEEE VR 2005 Workshop on New Directions in 3D User interfaces,*, pages 89–92, 2005.

[85] Raimund Dachselt and Anett Hbner. Three-dimensional menus: A survey and taxonomy. *Virtual Environments*, 31(1):5365, 2007.

[86] Andries van Dam, Andrew S. Forsberg, David H. Laidlaw, Joseph J. LaViola, and Rosemary M. Simpson. Immersive vr for scientific visualization: A progress report. *IEEE Computer Graphics and Applications*, 20:26–52, November 2000.

[87] Rudolph P. Darken, Terry Allard, and Lisa B. Achille. Spatial orientation and wayfinding in large-scale virtual spaces: An introduction. *Presence.*, 7(2):101–107, 1998.

[88] Raffaele de Amicis, Michele Fiorentino, and Andre Stork. Parametric interaction for cad application in virtual reality environment. In *International Conference on Design Tools and Methods in Industrial Engineering*, pages D3/43–D3/52, 2001.

[89] Maria Cristina Ferreira de Oliveira and Haim Levkowitz. From visual data exploration to visual data mining: A survey. *Visualization and Computer Graphic*, vol. 9 no. 3:378–394, 2003.

[90] Michael F. Deering. Holosketch: a virtual reality sketching/animation tool. *ACM Transactions on Computer-Human Interaction*, 2(3):220–238, 1995.

[91] Dan Diaper. *The Handbook of Task Analysis for Human-Computer Interaction*, chapter Task Analysis for Human-Computer Interaction, pages 5– 5 – 47. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2003.

[92] Guozhu Dong and Jinyan Li. Interestingness of discovered association rules in terms of neighborhood-based unexpectedness. In *Proceedings of the Second Pacific-Asia Conference on Research and Development in Knowledge Discovery and Data Mining*, PAKDD '98, pages 72–86, London, UK, UK, 1998. Springer-Verlag.

[93] Cdric Dumas, Patricia Plnacoste, and Christophe Chaillou. Dfinition d'un modle d'interaction pour une interface de travail tridimensionnelle partir d'exprimentations. In *HM'99*, 1999.

[94] H. Eidenberger. Visual data mining. In J. R. Smith, T. Zhang, & S. Panchanathan, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 5601 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 121–132, October 2004.

[95] Katja Einsfeld, S. Agne, M. Deller, A. Ebert, B. Klein, and C. Reuschling. Dynamic visualization and navigation of semantic virtual environments. In *Proceedings of the conference on Information Visualization (IV '06)*, pages 569–574. IEEE Computer Society, 2006.

[96] Katja Einsfeld, Achim Ebert, and Jurgen Wolle. Hannah: A vivid and flexible 3d information visualization framework. In *Proceedings of the 11th International Conference Information Visualization (IV '07)*, pages 720–725. IEEE Computer Society, 2007.

[97] Stephen R. Ellis. What are virtual environments? *IEEE Computer Graphics and Applications*, 14:17–22, 1994.

[98] T. Todd Elvins, David R. Nadeau, Rina Schul, and David Kirsh. Worldlets: 3d thumbnails for 3d browsing. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '98, pages 163–170, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

[99] Gurdal Ertek and Ayhan Demiriz. A framework for visualizing association mining results. In *Proceedings of the 21st international conference on Computer and Information Sciences*, ISCIS 2006, pages 593–602. Springer-Verlag, 2006.

[100] Usama M. Fayyad. Data mining and knowledge discovery in databases: Implications for scientific databases. In *Proceedings of the Ninth International Conference on Scientific and Statistical Database Management*, SSDBM '97, pages 2–11, Washington, DC, USA, 1997. IEEE Computer Society.

[101] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.

[102] Steven Feiner, Blair MacIntyre, Marcus Haupt, and Eliot Solomon. Windows on the world: 2d windows for 3d augmented reality. In *Proceedings of the 6th annual ACM symposium on User interface software and technology*, UIST '93, pages 145–155, New York, NY, USA, 1993. ACM.

[103] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer graphics : principles and practice*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1996.

[104] Andrew Forsberg, Kenneth Herndon, and Robert Zeleznik. Aperture based selection for immersive virtual environments. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, UIST '96, pages 95–96, New York, NY, USA, 1996. ACM.

[105] William J. Frawley, Gregory Piatetsky-Shapiro, and Christopher J. Matheus. Knowledge discovery in databases: An overview. *AI Magazine*, 13, n 3:57–70, 1992.

[106] Scott Frees and G. Drew Kessler. Precise and rapid interaction through scaled manipulation in immersive virtual environments. In *Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality*, VR '05, pages 99–106, Washington, DC, USA, 2005. IEEE Computer Society.

[107] A.A. Freitas. On rule interestingness measures. *Knowledge-Based Systems*, 12(5-6):309–315, 1999.

[108] Alex A. Freitas. Understanding the crucial role of attribute interaction in data mining. *Artificial Intelligence Review*, 16(3):177–199, 2001.

[109] Alex Alves Freitas. On objective measures of rule surprisingness. In *Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*, PKDD '98, pages 1–9, London, UK, 1998. Springer-Verlag.

[110] Michael Friendly. *Visualizing Categorical Data*. SAS Publishing, 1st edition, 2001.

[111] Yongjian Fu and Jiawei Hah. Meta-rule-guided mining of association rules in relational databases. In *Proceedings of the 1st International Workshop on Integration of Knowledge Discovery with Deductive and Object-Oriented Databases*, KDOOD'95, pages 39–46, 1995.

[112] P. Fuchs, B. Arnaldi, and J. Tisseau. *La ralit virtuelle et ses applications*. Presses des Mines, 2003.

[113] Philippe Fuchs, Sabine Coquillart, and Jean-Marie Burkhardt. *Le trait de la ralit virtuelle: Interfaage, immersion et interaction en environnement virtuel*. Presses des Mines, 2006.

[114] Takeshi Fukuda, Yasuhiko Morimoto, Shinichi Morishita, and Takeshi Tokuyama. Data mining with optimized two-dimensional association rules. *ACM Transactions on Database Systems (TODS)*, 26:179–213, June 2001.

[115] Peter Fule and John F. Roddick. Experiences in building a tool for navigating association rule result sets. In *Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation*, volume 32, pages 103 – 108, 2004.

[116] G. W. Furnas. Generalized fisheye views. In *CHI '86: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 16–23. ACM Press, 1986.

[117] Liqiang Geng and Howard J. Hamilton. Interestingness measures for data mining: A survey. *ACM Computing Surveys (CSUR)*, 38, September 2006.

[118] D. Gerber and D. Bechmann. The spin menu: A menu system for virtual environments. In *Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality*, VR '05, pages 271–272, Washington, DC, USA, 2005. IEEE Computer Society.

[119] Dominique Gerber. *Interaction 3D sur le plan de travail virtuel: Application aux dformations de forme libre*. PhD thesis, Universit Louis Pasteur de Strasbourg, 2004.

[120] James J. Gibson. *The Ecological Approach To Visual Perception*. Psychology Press, 1986.

[121] Bart Goethals and Jan Van Den Bussche. A priori versus a posteriori filtering of association rules. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery.*, pages 1–6, 1999.

[122] Bart Goethals and Mohammed J. Zaki. Advances in frequent itemset mining implementations. *SIGKDD Explorations*, 6(1):109–117, Jun 2003.

[123] Karam Gouda and Mohammed Javeed Zaki. Efficiently mining maximal frequent itemsets. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 163–170, Washington, DC, USA, 2001. IEEE Computer Society.

[124] Gosta Grahne and Jianfei Zhu. Fast algorithms for frequent itemset mining using fp-trees. *IEEE Transactions on Knowledge and Data Engineering*, 17(10):1347–1362, 2005.

[125] Saul Greenberg. *The Handbook of Task Analysis for Human-Computer Interaction*, chapter Working Through Task-Centered System Design, pages 49–65. Prentice Hall PTR, 2003.

[126] Nicolas Greffard, Fabien Picarougne, and Pascale Kuntz. Immersive dynamic visualization of interactions in a social network. In *GfKl*, pages 255–262, 2011.

[127] Jerome Grosjean, Jean-Marie Burkhardt, Sabine Coquillart, and Paul Richard. Evaluation of the command and control cube. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, ICMI '02, pages 473–, Washington, DC, USA, 2002. IEEE Computer Society.

[128] Jrme Grosjean and Sabine Coquillart. Command & control cube : a shortcut paradigm for virtual environments. In *7th EG Workshop on Virtual Environments & 5th Immersive Projection Technology Workshop*, 2001.

[129] Markus Gross. *Visual computing : the integration of computer graphics, visual perception and imaging*. Springer-Verlag, 1994.

[130] Timo Gtzelmann, Knut Hartmann, Andreas Nrnberger, and Thomas Strothotte. 3d spatial data mining on document sets for the discovery of failure causes in complex technical devices. In *Proceedings of the I2nd Int. Conf. on Computer Graphics Theory and Applications*, 2007.

[131] Fabrice Guillet and Howard J. Hamilton, editors. *Quality Measures in Data Mining*. Springer, 2007.

[132] Martin Hachet. *Interaction avec des Environnements Virtuels affichés au moyen d'Interfaces de Visualisation Collective*. PhD thesis, Université de Bordeaux 1, 2003.

[133] Jiawei Hah, Yongjian Fu, Wei Wang, Krzysztof Koperski, and Osmar Zaiane. Dmql: A data mining query language for relational databases. In *Proceedings of the SIGMOD Worshop on research issues on Data Mining and Knowledge Discovery*, pages 27–33. ACM, 1996.

[134] Michael Hahsler, Christian Buchta, and Kurt Hornik. Selective association rule generation. *Computational Statistics revue*, pages 303–315, March 2007.

[135] Jiawei Han and Yongjian Fu. Discovery of multiple-level association rules from large databases. In *Proceedings of the 21th International Conference on Very Large Data Bases*, VLDB '95, pages 420–431, 1995.

[136] Jiawei Han and Jian Pei. Mining frequent patterns by pattern-growth: methodology and implications. *ACM SIGKDD Explorations Newsletter, Special issue on Scalable data mining algorithms*, 2(2):14–20, 2000.

[137] Chris Hand. A survey of 3d interaction techniques. *Computer Graphics Forum*, 16(5):269281, 1997.

[138] Julie M. Harris. Binocular vision: moving closer to reality. *Philosophical transactions - Royal Society. Mathematical, physical and engineering sciences*, 362(1825):2721–2739, 2004.

[139] M. Hascot and M. Beaudoin-Lafon. Visualisation interactive dinformation. *Information, Interaction, Intelligence*, 1:1, 2001.

[140] Michele Heim. *Cyberspace/cyberbodies/cyberpunk: cultures of technological embodiment*, chapter The design of virtual reality, pages 65–78. 1995.

[141] R. J. Hendley, N. S. Drew, A. M. Wood, and R. Beale. Narcissus: visualising information. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS '95)*, pages 90–96. Morgan Kaufmann Publishers Inc., 1999.

[142] Ivan Herman, Ieee Computer Society, Guy Melancon, and M. Scott Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6:24–43, 2000.

[143] W. Hibbard, H. Levkowitz, J. Haswell, P. Rheingans, and F. Schroeder. Interaction in perceptually-based visualization. *Perceptual Issues in Visualization*, IFIP Series on Computer Graphics:23–32, 1995.

[144] Robert J. Hilderman and Howard J. Hamilton. Heuristic measures of interestingness. In *Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '99, pages 232–241, London, UK, 1999. Springer-Verlag.

[145] Jochen Hipp and Ulrich Güntzer. Is pushing constraints deeply into the mining algorithms really what we want?: an alternative approach for association rule mining. *ACM SIGKDD Explorations Newsletter*, 4:50–55, June 2002.

[146] Heike Hofmann, Arno P. J. M. Siebes, and Adalbert F. X. Wilhelm. Visualizing association rules with interactive mosaic plots. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 227–235, New York, NY, USA, 2000. ACM.

[147] John H. Holland, Keith J. Holyoak, Richard E. Nisbett, and Paul R. Thagard. *Induction: processes of inference, learning, and discovery.* MIT Press, Cambridge, MA, USA, 1986.

[148] Kian huat Ong, Kok leong Ong, Wee keong Ng, and Ee peng Lim. Crystalclear: active visualization of association rules. In *International Workshop on Active Mining*, AM2002, 2002.

[149] Geoffrey S. Hubona and Gregory W. Shirah. *Ambient Intelligence For Scientific Discovery*, volume 3345/2005, chapter Spatial Cues in 3D Visualization, pages 104–128. Spinger, 2005.

[150] Takeo Igarashi, Rieko Kadobayashi, Kenji Mase, and Hidehiko Tanaka. Path drawing for 3d walkthrough. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, UIST '98, pages 173–174, New York, NY, USA, 1998. ACM.

[151] Tomasz Imielinski and Heikki Mannila. A database perspective on knowledge discovery. *Communications of the ACM*, 39:58–64, November 1996.

[152] Tomasz Imielinski and Aashu Virmani. Association rules... and what's next? – towards second generation data mining systems. *Advances in Databases and Information Systems*, 1475:6, 1998.

[153] Tomasz Imieliński and Aashu Virmani. Msql: A query language for database mining. *Data Mining and Knowledge Discovery*, 3:373–408, December 1999.

[154] Franois-Xavier Inglese, Philippe Lucidarme, Paul Richard, and Jean-Louis Ferrier. Previse - a human-scale virtual environment with haptic feedback. In *Second International Conference on Informatics in Control, automation and Robotics*, pages 140–145. INSTICC Press, 2005.

[155] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st conference on Visualization*, VIS '90, pages 361–378. IEEE Computer Society Press, 1990.

[156] Jerry Isdale. *What is virtual reality ?* a homebrew introduction and information resource list, 1993.

[157] Richard H. Jacoby and Stephen R. Ellis. Using virtual menus in a virtual environment. *SPIE*, 1668:39–48, 1992.

[158] Richard H. Jacoby, Mark Ferneau, and Jim Humphries. Hands-off interaction with menus in virtual spaces. *SPIE*, 2177:355–371, 1994.

[159] T. J. Jankun-Kelly, Kwan-Liu Ma, and Michael Gertz. A model and framework for visualization exploration. *IEEE Transactions on Visualization and Computer Graphics*, 13:357–369, March 2007.

[160] Baptiste Jeudy and Jean-François Boulicaut. Optimization of association rule mining queries. *Intelligent Data Analysis*, 6:341–357, September 2002.

[161] Brian Johnson and Ben Shneiderman. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the 2nd conference on Visualization (VIS '91)*, pages 284–291. IEEE Computer Society Press, 1991.

[162] A. Kadri, A. Lecuyer, and J.-M. Burkhardt. The visual appearance of user's avatar can influence the manipulation of both real devices and virtual objects. In *IEEE Symposium on 3D User Interfaces*, 2007.

[163] A. Kadri, A. Lecuyer, J.-M. Burkhardt, and S. Richir. The influence of visual appearance of user's avatar on the manipulation of objects in virtual environments. In *IEEE Symposium on 3D User Interfaces*, pages 291 – 292, 2007.

[164] R S Kalawsky. Exploiting virtual reality techniques in education and training: Technological issues. *AGOCGSIMA Report Series*, 26(1):1356–5370, 1996.

[165] Roy Kalawsky and Graeme Simpkin. Automating the display of third person/stealth views of virtual environments. *Presence: Teleoper. Virtual Environ.*, 15(6):717–739, 2006.

[166] L. N. Kalisperis, G. Otto, K. Muramoto, J. S. Gundrum, R. Masters, and B Orland. An affordable immersive environment in beginning design studio education. In *ACADIA 2002, Thresholds Between Real and Virtual: Design Research, Education, and Practice in the Space Between the Physical and the Virtual,*, pages 49–56, 2002.

[167] D.A. Keim and H.-P.; Kriegel. Visualization techniques for mining large databases: a comparison. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):923 – 938, 1996.

[168] Daniel A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[169] Yoshifumi Kitamura, Tomohiko Higashi, Toshihiro Masaki, and Fumio Kishino. Virtual chopsticks: Object manipulation using multiple exact interactions. In *Proceedings of the IEEE Virtual Reality*, VR '99, pages 198–, Washington, DC, USA, 1999. IEEE Computer Society.

[170] Mika Klemettinen, Heikki Mannila, Pirjo Ronkainen, Hannu Toivonen, and A. Inkeri Verkamo. Finding interesting rules from large sets of discovered association rules. In *Proceedings of the third international conference on Information and knowledge management*, CIKM '94:, pages 401–407, New York, NY, USA, 1994. ACM.

[171] Ioannis Kopanakis and Babis Theodoulidis. Visual data mining modeling techniques for the visualization of mining outcomes. *Journal of Visual Languages & Computing*, 14(6):543–589, 2003.

[172] Uwe Krohn. Vineta: navigation through virtual information spaces. In *Proceedings of the workshop on Advanced visual interfaces(AVI '96)*, pages 49–58, New York, NY, USA, 1996. ACM.

[173] John Krygier and Denis Wood. *Making Maps: A Visual Guide to Map Design for GIS*. The Guilford Press, 2005.

[174] Pascale Kuntz, Fabrice Guillet, Rémi Lehn, and Henri Briand. A user-driven process for mining association rules. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '00, pages 483–489. Springer-Verlag, 2000.

[175] Gordon Kurtenbach and William Buxton. User learning and performance with marking menus. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, CHI '94, pages 258–264, New York, NY, USA, 1994. ACM.

[176] Celine Latulipe, Craig S. Kaplan, and Charles L. A. Clarke. Bimanual and unimanual image alignment: an evaluation of mouse-based techniques. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, UIST '05, pages 123–131, New York, NY, USA, 2005. ACM.

[177] Joseph J. Laviola. Msvt: A multimodal scientific visualization tool. In *the 3ed IASTED International Conference on Computer Graphics and Imaging*, pages 1–17, 2000.

[178] Anatole Lecuyer, Christine Megard, Jean-Marie Burkhardt, Eads Ccr, Taegi Lim, Philippe Coiffet, Ludovic Graux, and Sabine Coquillart. The effect of haptic, visual and auditory feedback on an insertion task on a 2-screen workbench. In *In Proceedings of the Immersive Projection Technology Symposium*, 2002.

[179] Adrien Marie Legendre. *Nouvelles mthodes pour la dtermination des orbites des comtes*. 1805.

[180] Carson Kai-Sang Leung, Laks V. S. Lakshmanan, and Raymond T. Ng. Exploiting succinct constraints using fp-trees. *ACM SIGKDD Explorations Newsletter*, 4:40–49, June 2002.

[181] Jiuyong Li. On optimal rule discovery. *IEEE Transactions on Knowledge and Data Engineering*, 18, 2006.

[182] Wenmin Li, Jiawei Han, and Jian Pei. Cmar: accurate and efficient classification based on multiple class-association rules. In *Proceedings IEEE International Conference on Data Mining*, pages 369– 376, 2001.

[183] Jiandong Liang and Mark Green. Jdcad: A highly interactive 3d modeling system. *In Computer and Graphics*, 18(4):499–506, 1994.

[184] Bing Liu, Wynne Hsu, Lai-Fun Mun, and Hing-Yan Lee. Finding interesting patterns using user expectations. *IEEE Transactions on Knowledge and Data Engineering*, 11(6):817–832, 1999.

[185] Yan Liu and Gavriel Salvendy. Design and evaluation of visualization support to facilitate association rules modeling. *International Journal of Human-Computer Interaction*, 21(1):15–38, 2006.

[186] Jianxiong Luo and Susan M Bridges. Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection. *International Journal of Intelligent Systems*, 15(8):687–703, 2000.

[187] Yiming Ma, Bing Liu, and Ching Kian Wong. Web for data mining: organizing and interpreting the discovered rules using the web. *ACM SIGKDD Explorations Newsletter*, 2:16–23, June 2000.

[188] Alan M. MacEachren. *How Maps Work : Representation, Visualization, and Design*. Guilford Press, 1995.

[189] John A. Major and John J. Mangano. Selecting among rules induced from a hurricane database. *Journal of Intelligent Information Systems*, 4(1):39–52, 1995.

[190] Jonathan I. Maletic, Andrian Marcus, Greg Dunlap, and Jason Leigh. Visualizing object-oriented software in virtual reality. In *Proceedings of the Ninth International Workshop on Program Comprehension*, IWPC'01, pages 26–38. IEEE Computer Society, 2001.

[191] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3):259–289, 1997.

[192] Bianchi Serique Meiguins, Ricardo Melo, Casseb do Carmo, Leonardo Almeida, Aruanda Simoes Gonalves, Srgio Clayton V. Pinheiro, and Marcelo de Brito Garcia. Multidimensional information visualization using augmented reality. In *Proceedings of ACM international conference on Virtual reality continuum and its applications (VRCIA '06)*, 2006.

[193] Rosa Meo, Giuseppe Psaila, and Stefano Ceri. An extension to sql for mining association rules. *Data Mining and Knowledge Discovery*, 2:195–224, June 1998.

[194] DR Mestre and P Fuchs. *Trait de la Ralit Virtuelle*, chapter Immersion et Prsence, pages 309–338. Ecole des Mines de Paris, 2006.

[195] Ryszard S. Michalski, Ivan Bratko, and Avan Bratko. *Machine Learning and Data Mining; Methods and Applications*. John Wiley & Sons, Inc., 1998.

[196] P. Milgram, D. Drascic, J. J. Grodski, A. Restogi, S. Zhai, and C. Zhou. Merging real and virtual worlds. In *IMAGINA95*, pages 218–30, 1995.

[197] Paul MILGRAM and Fumio KISHINO. A taxonomy of mixed reality visual displays. *IEICE transactions on information and systems*, E77(12):1321–1329, 1994.

[198] Mark R. Mine. Isaac : A virtual environment tool for the interactive construction of virtual worlds. Technical report, University of North Carolina, 1995.

[199] Mark R. Mine, Frederick P. Brooks, Jr., and Carlo H. Sequin. Moving objects in space: exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 19–26, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[200] Henry Montgomery. Decision rules and the search for a dominance structure: Towards a process model of decision making*. *Advances in Psychology*, 14:343369, 1983.

[201] Lukas Mroz and Helwig Hauser. Rtvr: a flexible java library for interactive volume rendering. In *VISUALIZATION'01 : Proceedings of the Conference on Visualization*, pages 279–286. IEEE Computer Society Press, 2001.

[202] Edward J. Mulrow. The visual display of quantitative information. *Technometrics*, 44(4):400, 2002.

[203] Henrik R. Nagel, Erik Granum, Søren Bovbjerg, and Michael Vittrup. Immersive visual data mining: The 3dvdm approach. *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*, pages 281–311, 2008.

[204] Henrik R Nagel, Erik Granum, and Peter Musaeus. Methods for visual mining of data in virtual reality. In *Proceedings of the International Workshop on Visual Data Mining in conjunction with 2nd European Conference on Machine Learning and 5th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 13–27, 2001.

[205] David Nahon. Salles immersives et cubes de ralit virtuelle, une premire mondiale sur pc : le sas cube. In *Imagina*, 2002.

[206] L. Nelson, D. Cook, and C. Cruz-Neira. Xgobi vs the c2: Results of an experiment comparing data visualization in a 3-d immersive virtual reality environement with a 2-d workstation display. *computational statistics 14*, pages 39–51, 1999.

[207] Raymond T. Ng, Laks V. S. Lakshmanan, Jiawei Han, and Alex Pang. Exploratory mining and pruning optimizations of constrained associations rules. *ACM SIGMOD Record*, 27:13–24, June 1998.

[208] Oliver Niggemann. *Visual Data Mining of Graph-Based Data.* PhD thesis, University of Paderborn, Jun 2001.

[209] Pang ning Tan and Vipin Kumar. Interestingness measures for association patterns: A perspective. Technical report, Departement of Computer Science and engineering, University of Minnesota, 2000.

[210] Tetsuro Ogi, Yoshisuke Tateyama, and So Sato. Visual data mining in immersive virtual environment based on 4k stereo images. In *VMR '09: Proceedings of the 3rd International Conference on Virtual and Mixed Reality*, pages 472–481. Springer-Verlag, 2009.

[211] Miho Ohsaki, Shinya Kitaguchi, Kazuya Okamoto, Hideto Yokoi, and Takahira Yamaguchi. Evaluation of rule interestingness measures with a clinical dataset on hepatitis. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD '04, pages 362–373, New York, NY, USA, 2004. Springer-Verlag New York, Inc.

[212] A. Olwal and S Feiner. The flexible pointer: An interaction technique for augmented and virtual reality. In *Proceedings of ACM Symposium on User Interface Software and Technology*, number 81-82, 2003.

[213] M. Ortega and S. Coquillart. Prop-based haptic interaction with co-location and immersion: an automotive application. In *IEEE International Workshop on Haptic Audio Visual Environments and their Applications*, 2005.

[214] K. Osawa, N. Asai, M. Suzuki, Y. Sugimoto, and F. Saito. An immersive programming system: Ougi. In *Proceedings of the 12th International Conference on Artificial Reality and Telexistence (ICAT2002)*, pages 36–43, 2002.

[215] Alexis Paljic, Sabine Coquillart, Jean-Marie Burkhardt, and Paul Richard. A study of distance of manipulation on the responsive workbench. In *Immersive Projection Technology*, IPT2002, 2002.

[216] G. Parker, G. Franck, and C. Ware. Visualization of large nested graphs in 3d: Navigation and interaction. *Journal of Visual Languages & Computing*, 9:299–317, 1998.

[217] Nicolas Pasquier, Rafik Taouil, Yves Bastide, Gerd Stumme, and Lotfi Lakhal. Generating a condensed representation for association rules. *Journal of Intelligent Information Systems*, 24(1):29–60, 2005.

[218] Jian Pei and Jiawei Han. Can we push more constraints into frequent pattern mining? In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 350–354, New York, NY, USA, 2000. ACM.

[219] G. Piatetsky-Shapiro. *Knowledge Discovery in Databases*, chapter Discovery, Analysis, and Presentation of Strong Rules, page 229248. AAAI/MIT Press, 1991.

[220] G.G. Pickett, R.M. Grinstein. Iconographic displays for visualizing multidimensional data. *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, pages 514–519, 1988.

[221] Jeffrey S. Pierce, Andrew S. Forsberg, Matthew J. Conway, Seung Hong, Robert C. Zeleznik, and Mark R. Mine. Image plane interaction techniques in 3d immersive environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, I3D '97, pages 39–ff., New York, NY, USA, 1997. ACM.

[222] Jeffrey S. Pierce, Brian C. Stearns, and Randy Pausch. Voodoo dolls: seamless interaction at multiple scales in virtual environments. In *Proceedings of the 1999 symposium on Interactive 3D graphics*, I3D '99, pages 141–145, New York, NY, USA, 1999. ACM.

[223] Ben Piper, Carlo Ratti, and Hiroshi Ishii. Illuminating clay: a 3-d tangible interface for landscape analysis. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, CHI '02, pages 355–362, New York, NY, USA, 2002. ACM.

[224] Catherine Plaisant, Jesse Grosjean, and Benjamin B. Bederson. Spacetree: Supporting exploration in large node link tree, design evolution and empirical evaluation. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, pages 57–64. IEEE Computer Society, 2002.

[225] François Poulet and Thanh Nghi Do. *Visual Data Mining*, chapter Interactive Decision Tree Construction for Interval and Taxonomical Data, pages 123–135. Springer-Verlag, 2008.

[226] I. Poupyrev, T. Ichikawa, S. Weghorst, and M. Billinghurst. Egocentric object manipulation in virtual environments: Empirical evaluation of interaction techniques. *Computer Graphics Forum*, 17:4152, 1998.

[227] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in vr. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, UIST '96, pages 79–80, New York, NY, USA, 1996. ACM.

[228] IVAN POUPYREV and TADAO Ichikawa. Manipulating objects in virtual worlds: Categorization and empirical evaluation of interaction techniques. *Journal of Visual Languages & Computing*, 10(1):1935, 1999.

[229] Foster John Provost and John M. Aronis. Scaling up inductive learning with massive parallelism. *Machine Learning*, 3(1):33–46, 1996.

[230] Andy Prykeand and Russell Beale. Interactive comprehensible data mining. *Ambient Intelligence for Scientific Discovery*, 3345:48–65, 2005.

[231] GWM Rauterberg, M Fjeld, Bichsel Krueger, H, U M, Leonhardt, and Markus Meier. Build-it : a video-based interaction technique for a planning tool for construction and design. In *Work With Display Units (WWDU),*, page 175176, 1997.

[232] Jun Rekimoto. Pick-and-drop: a direct manipulation technique for multiple computer environments. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, UIST '97, pages 31–39, New York, NY, USA, 1997. ACM.

[233] Howard Rheingold. *Virtual Reality.* Summit Books, 1991.

[234] Paul Richard. *Analyse de linteraction homme-monde virtuel lors de tches de manipulation dobjets dformables.* PhD thesis, Universit Pierre et Marie Curie (Paris 6), 1996.

[235] William S. Cleveland Richard A. Becker and Allan R. Wilks. Dynamic graphics for data analysis. *Statistical Science*, 2(4):355–383., 1987.

[236] George Robertson, Mary Czerwinski, Kevin Larson, Daniel C. Robbins, David Thiel, and Maarten van Dantzich. Data mountain: using spatial memory for document management. In *Proceedings of the 11th annual ACM symposium on User interface software and technology (UIST '98)*, pages 153–162, 1998.

[237] George G Robertson, Jock D Mackinlay, and Stuart K. Card. Cone trees: animated 3d visualizations of hierarchical information. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, pages 189 – 194, 1991.

[238] Warren Robinett and Richard Holloway. Implementation of flying, scaling and grabbing in virtual worlds. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, I3D '92, pages 189–192, New York, NY, USA, 1992. ACM.

[239] John F. Roddick and Sally Rice. What's interesting about cricket?: on thresholds and anticipation in discovered rules. *ACM SIGKDD Explorations Newsletter*, 3:1–5, July 2001.

[240] ELTING L. S., MARTIN C. G., ; CANTOR S. B., and ; RUBENSTEIN E. B. nfluence of data display formats on physician investigators' decisions to stop clinical trials : prospective trial with repeated measures. *British Medical Journal*, 318(7197):1527–1531, 1999.

[241] J Grosjean S Coquillart. Interaction 3d, paradigmes et mtaphores. *Trait de la ralit virtuelle*, 2003.

[242] Sigal Sahar. Interestingness via what is not interesting. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '99, pages 332–336, New York, NY, USA, 1999. ACM.

[243] Ashoka Savasere, Edward Omiecinski, and Shamkant B. Navathe. An efficient algorithm for mining association rules in large databases. In *Pceedings of the 21th International Conference on Very Large Data Bases*, pages 432–444, 1995.

[244] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings of IEEE Symposium on Visual Languages*, pages 336– 343, 1996.

[245] Ben Shneiderman. Why not make interfaces better than 3d reality? *IEEE Computer Graphics and Applications*, 23(6):12–15, 2003.

[246] Yedendra Babu Shrinivasan and Jarke J. van Wijk. Supporting the analytical reasoning process in information visualization. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 1237–1246, New York, NY, USA, 2008. ACM.

[247] Avi Silberschatz and Alexander Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *Proceedings of KDD-95*, pages 275 – 281, 1995.

[248] Avi Silberschatz and Alexander Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):970–974, December 1996.

[249] Avi Silberschatz and Alexander Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering*, 8:970–974, December 1996.

[250] Simeon J. Simoff, Michael H. Böhlen, and Arturas Mazeika. Visual data mining. chapter Visual Data Mining: An Introduction and Overview, pages 1–12. Springer-Verlag, Berlin, Heidelberg, 2008.

[251] Herbert A. Simon. *Models of thought*. Yale University Press, 1979.

[252] Mel Slater, John McCarthy, and Francesco Maringelli. The influence of body movement on subjective presence in virtual environments. *The Journal of the Human Factors and Ergonomics Society*, 40(3):469–477, 1998.

[253] Mel Slater, Martin Usoh, Steve Benford, Dave Snowdon, Chris Brown, Tom Rodden, Gareth Smith, and Sylvia Wilbur. Distributed extensible virtual reality laboratory (devrl): a project for co-operation in multi-participant environments. In *Proceedings of the Eurographics workshop on Virtual environments and scientific visualization*, pages 137–148, London, UK, 1996. Springer-Verlag.

[254] P. Smyth and R. M. Goodman. An information theoretic approach to rule induction from databases. *IEEE Transactions on Knowledge and Data Engineering*, 4(4):301–316, 1992.

[255] Ian Spence. Visual psychophysics of simple graphical elements. *Journal of Experimental Psychology: Human Perception and Performance.*, 16(4):683–692, 1990.

[256] R. Spence. *Information Visualization*. Addison-Wesley: Essex, England, 2001.

[257] Ramakrishnan Srikant and Rakesh Agrawal. Mining quantitative association rules in large relational tables. In *Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, SIGMOD '96, pages 1–12, New York, NY, USA, 1996. ACM.

[258] Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology*, pages 3–17, 1996.

[259] Ramakrishnan Srikant and Rakesh Agrawal. Mining generalized association rules. *Future Generation Computer Systems*, 13(2–3):161–180, 1997.

[260] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. Web usage mining: discovery and applications of usage patterns from web data. *ACM SIGKDD Explorations Newsletter*, 1:12–23, January 2000.

[261] Anthony Steed and Chris Parker. 3d selection strategies for head tracked and non-head tracked operation of spatially immersive displays. In *8th International Immersive Projection Technology Workshop*, pages 13–14, 2004.

[262] Frank Steinicke, Klaus H. Hinrichs, and Timo Ropinski. Virtual reflections and virtual shadows in mixed reality environments. In *Proceedings of the 10th International Conference on Human-Computer Interaction*, INTERACT'05, pages 1018–1021, Rome, 2005.

[263] L Sternberger. Deformable ray-casting interaction technique. In *IEEE Young Virtual Reality Conferences*, 2005.

[264] Richard Stoakley, Matthew J. Conway, and Y Pausch. Virtual reality on a wim: interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '95, pages 265–272, 1995.

[265] T.A. Stoffregen, B. Bardy, L.J. G. Smart, and R.J. Pagulayan. *Virtual and adaptative environments: Applications, implications, and Human performance issues*, chapter On the nature and evaluation of fidelity in virtual environments, pages 111–128. Mahwah, NJ: Lawrence Erlbaum., 2003.

[266] D. J. Sturman, D. Zeltzer, and S. Pieper. Hands-on interaction with virtual environments. In *Proceedings of the 2nd annual ACM SIGGRAPH symposium on User interface software and technology*, UIST '89, pages 19–24, New York, NY, USA, 1989. ACM.

[267] Zsolt Szalavri and Michael Gervautz. The personal interaction panel a two-handed interface for augmented reality. *Computer Graphics Forum*, 16(3):C335C346, 1997.

[268] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Selecting the right interestingness measure for association patterns. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 32–41. ACM, 2002.

[269] Vildan Tanriverdi and Robert J. K. Jacob. Interacting with eye movements in virtual environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '00, pages 265–272. ACM, 2000.

[270] Monica Tavanti and Mats Lind. 2d vs 3d, implications on spatial memory. In *Proceedings of the IEEE Symposium on Information Visualization 2001*, INFOVIS'01, page 139. IEEE Computer Society, 2001.

[271] Kesaraporn Techapichetvanich and Amitava Datta. Visar : A new technique for visualizing mined association rules. In *International conference on advanced data mining and applications*, ADMA 2005. Springer-Verlag, 2005.

[272] Tudor Teusan, Gilles Nachouki, Henri Briand, and Jacques Philippe. Discovering association rules in large, dense databases. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '00, pages 638–645. Springer-Verlag, 2000.

[273] Alfredo R. Teyseyre and Marcelo R. Campo. An overview of 3d software visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):87–105, 2009.

[274] Jaques Tisseau. *Realite virtuelle : autonomie in virtuo*. PhD thesis, University of Rennes1, 2001.

[275] H. Toivonen, M. Klemettinen, P. Ronkainen, K. Hatonen, and H. Mannila. Pruning and grouping of discovered association rules. *Workshop on Statistics, Machine Learning, and Knowledge Discovery in Databases*, pages 47–52, 1995.

[276] Melanie Tory, Arthur E. Kirkpatrick, M. Stella Atkins, and Torsten Moller. Visualization task performance with 2d, 3d, and combination displays. *IEEE Transactions on Visualization and Computer Graphics*, 12:2–13, January 2006.

[277] Melanie Tory and Torsten Moller. Rethinking visualization: A high-level taxonomy. In *Proceedings of IEEE Symposium on Information Visualization*, INFOVIS '04, pages 151–158. IEEE Computer Society, 2004.

[278] Edward R. Tufte. *Envisioning Information*. Graphics Press, 1990.

[279] Antony Unwin. Visualisation for data mining. In *International Conference on Data Mining, Visualization and Statistical System*, 2000.

[280] Antony Unwin, Heike Hofmann, and Klaus Bernt. The twokey plot for multiple association rules control. In *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '01, pages 472–483, 2001.

[281] Padhraic Smyth Usama Fayyad, Gregory Piatetsky-Shapiro. From data mining to knowledge discovery in databases. *AI MAGAZINE*, 17(3):37–54, 1996.

[282] Martin Usoh, Kevin Arthur, Mary C. Whitton, Rui Bastos, Anthony Steed, Mel Slater, and Frederick P. Brooks, Jr. Walking walking-in-place flying, in virtual environments. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 359–364, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[283] J. J. Valdes. Virtual reality representation of information systems and decision rules: An exploratory tool for understanding data and knowledge. In *the 9-th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, 2003.

[284] J.J. van Ham, F. van Wijk. Beamtrees: compact visualization of large hierarchies. *Proceedings of IEEE Symposium on Information Visualization (INFOVIS '02)*, pages 93– 100, 2002.

[285] Weixin Wang, Hui Wang, Guozhong Dai, and Hongan Wang. Visualization of large hierarchical data by circle packing. In *Proceedings of SIGCHI conference on Human Factors in computing systems (CHI '06)*, pages 517–520. ACM, 2006.

[286] Matthew Ward, Georges Grinstein, and Daniel Keim. *Interactive Data Visualization: Foundations, Techniques, and Applications*. A. K. Peters, Ltd., 2010.

[287] Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[288] Colin Ware and Glenn Franck. Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Transactions on Graphics (TOG)*, 15(2):121–140, 1996.

[289] Colin Ware and Steven Osborne. Exploration and virtual camera control in virtual three dimensional environments. *SIGGRAPH Computer Graphics*, 24:175–183, 1990.

[290] Gerold Wesche and Marc Droske. Conceptual free-form styling on the responsive workbench. In *Proceedings of the ACM symposium on Virtual reality software and technology*, VRST '00, pages 83–91, New York, NY, USA, 2000. ACM.

[291] Leland Wilkinson. *The Grammar of Graphics (Statistics and Computing)*. Springer-Verlag, 2005.

[292] Remco Chang William A Pike, John Stasko and Theresa A O'Connell. The science of interaction. *Information Visualization*, 8:263–274, 2009.

[293] Christopher Williamson and Ben Shneiderman. The dynamic homefinder: evaluating dynamic queries in a real-estate information exploration system. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '92, pages 338–346. ACM Press, 1992.

[294] U. Wiss and D.A. Carr. An empirical study of task support in 3d information visualizations. In *IEEE International Conference on Information Visualization*, pages 392 – 399, 1999.

[295] Pak Chung Wong. Visual data mining. *IEEE Computer Graphics and Applications*, 19(5):20–21, 1999.

[296] Pak Chung Wong and R. Daniel Bergeron. 30 years of multidimensional multivariate visualization. In *Scientific Visualization, Overviews, Methodologies, and Techniques*, pages 3–33, Washington, DC, USA, 1997. IEEE Computer Society.

[297] Pak Chung Wong, Paul Whitney, and Jim Thomas. Visualizing association rules for text mining. In *Proceedings of the 1999 IEEE Symposium on Information Visualization*, pages 120–123, 1999.

[298] Yue Xu and Yuefeng Li. Generating concise association rules. In *Proceedings of the sixteenth ACM conference on Conference on Information and Knowledge Management*, pages 781–790. ACM, 2007.

[299] Claudio Haruo Yamamoto, Magaly Lika Fujimoto, and Solange Oliveira Rezende. An itemset-driven cluster-oriented approach to extract compact and meaningful sets of association rules. In *Proceedings of the Sixth International Conference on Machine Learning and Applications*, IV '07, pages 87–92, Washington, DC, USA, 2007. IEEE Computer Society.

[300] Li Yang. Visualizing frequent itemsets, association rules, and sequential patterns in parallel coordinates. In *Proceedings of the 2003 international conference on Computational science and its applications*, ICCSA'03, pages 21–30. Springer-Verlag, 2003.

[301] Y. Y. Yao, S. K. Michael Wong, and Cory J. Butz. On information-theoretic measures of attribute importance. In *Proceedings of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining*, PAKDD '99, pages 133–137, London, UK, 1999. Springer-Verlag.

[302] Ji Soo Yi, Youn ah Kang, John Stasko, and Julie Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE*

Transactions on Visualization and Computer Graphics, 13:1224–1231, November 2007.

[303] Yan Allison York, L. Yan, R. S. Allison, and S. K. Rushton. New simple virtual walking method – walking on the spot. In *In 8 th Annual Immersive Projection Technology (IPT) Symposium Electronic Proceedings*, 2004.

[304] Mohammed Zaki. Mining non-redundant association rules. *Data Mining and Knowledge Discovery*, 9:223–248, 2004.

[305] Mohammed J. Zaki. Generating non-redundant association rules. *International Conference on Knowledge Discovery and Data Mining*, pages 34 – 43, 2000.

[306] Mohammed J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42:31–60, January 2001.

[307] David Zeltzer. Autonomy, interaction, and presence. *Presence: Teleoper. Virtual Environ.*, 1:127–132, 1992.

[308] Shumin Zhai, William Buxton, and Paul Milgram. The silk cursor: investigating transparency for 3d target acquisition. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, CHI '94, pages 459–464, New York, NY, USA, 1994. ACM.

[309] Kaidi Zhao and Bing Liu. Opportunity map: A visualization framework for fast identification of actionable knowledge. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 60–67, 2005.

[310] Qiankun Zhao and Sourav S. Bhowmick. Association rule mining: A survey. Technical report, CAIS, Nanyang Technological University, Singapore, 2003.

l'université nantes angers le mans
PÔLE DE RECHERCHE ET D'ENSEIGNEMENT SUPÉRIEUR

UNIVERSITÉ DE NANTES

Thèse de Doctorat

Zohra Ben Said - Guefrech

A virtual reality-based approach for interactive and visual mining of association rules

Résumé

Cette thèse se situe à l'intersection de deux domaines actifs de recherche : la fouille de règles d'association et la réalité virtuelle. Les limites majeures des algorithmes d'extraction de règles d'association sont (i) la grande quantité de règles produites et (ii) leur faible qualité. Dans la littérature, plusieurs solutions ont été proposées pour remédier à ce problème, comme le post-traitement de règles d'association qui permet la validation des règles et l'extraction de connaissances utiles. Cependant, alors que les règles sont extraites automatiquement par des algorithmes combinatoires, le post-traitement de règles est effectué par l'utilisateur. La visualisation peut aider l'utilisateur à faire face à une grande quantité de règles en les représentants sous forme visuelle. Afin de trouver les connaissances pertinentes dans les représentations visuelles, l'utilisateur doit interagir avec la représentation de règles d'association. Par conséquent, il est essentiel de fournir à l'utilisateur des techniques d'interaction efficaces.

Ce travail aborde deux problèmes essentiels : la représentation de règles d'association afin de permettre à l'utilisateur de détecter très rapidement les règles les plus intéressantes et l'exploration interactive des règles. Le premier exige une métaphore intuitive de représentation de règles d'association. Le second nécessite un processus d'exploration très interactif permettant à l'utilisateur de fouiller l'espace de règles en se concentrant sur les règles intéressantes.

Les principales contributions de ce travail peuvent être résumées comme suit :

- Nous proposons une nouvelle classification pour les techniques de fouille visuelles de données, basée sur des représentations en 3D et des techniques d'interaction. Une telle classification aide l'utilisateur à choisir une configuration pertinente pour son application.
- Nous proposons une nouvelle métaphore de visualisation pour les règles d'association qui prend en compte les attributs de la règle, la contribution de chacun d'eux et leurs corrélations.
- Nous proposons une méthodologie pour l'exploration interactive de règles d'association. Elle est conçue pour faciliter la tâche de l'utilisateur face à des grands ensembles de règles en tenant en compte ses capacités cognitives. Dans cette méthodologie, des algorithmes locaux sont utilisés pour recommander les meilleures règles basées sur une règle de référence proposée par l'utilisateur. Ensuite, l'utilisateur peut à la fois diriger l'extraction et le post-traitement des règles en utilisant des opérateurs d'interaction appropriés.
- Nous avons développé un outil qui implémente toutes les fonctionnalités de la méthodologie. Notre outil est basé sur un affichage intuitif dans un environnement virtuel et prend en charge plusieurs méthodes d'interaction.

Mots clés

Règles d'association, Réalité virtuelle, fouille visuelle de données, Visualisation, Exploration Interactive de Règles.

Abstract

This thesis is at the intersection of two active research areas : Association Rules Mining and Virtual Reality.

The main limitations of the association rule extraction algorithms are (i) the large amount of the generated rules and (ii) their low quality. Several solutions have been proposed to address this problem such as, the post-processing of association rules that allows rule validation and extraction of useful knowledge. Whereas rules are automatically extracted by combinatorial algorithms, rule post-processing is done by the user. Visualisation can help the user facing the large amount of rules by representing them in visual form. In order to find relevant knowledge in visual representations, the user needs to interact with these representations. To this aim, it is essential to provide the user with efficient interaction techniques. This work addresses two main issues : an association rule representation that allows the user quickly detection of the most interesting rules and interactive exploration of rules. The first issue requires an intuitive representation metaphor of association rules. The second requires an interactive exploration process allowing the user to explore the rule search space focusing on interesting rules.

The main contributions of this work can be summarised as follows :

- We propose a new classification for Visual Data Mining techniques, based on both 3D representations and interaction techniques. Such a classification helps the user choosing a visual representation and an interaction technique for his/her application.
- We propose a new visualisation metaphor for association rules that takes into account the attributes of the rule, the contribution of each one, and their correlations.
- We propose a methodology for interactive exploration of association rules to facilitate the user task facing large sets of rules taking into account his/her cognitive capabilities. In this methodology, local algorithms are used to recommend better rules based on a reference rule which is proposed by the user. Then, the user can both drives extraction and post-processing of rules using appropriate interaction operators.
- We developed a tool that implements all the methodology functionality. The tool is based on an intuitive display in a virtual environment and supports multiple interaction methods.

Key Words

Association Rules Mining, Virtual Reality, Visualisation, Visual Data Mining, Interactive Rules Exploration.