Browsing Large Concept Lattices through Tree Extraction and Reduction Methods
Cassio Melo¹, Bénédicte Le-Grand² and Marie-Aude Aufaure¹

¹École Centrale Paris, France
²Université Paris 1 Panthéon – Sorbonne, France

ABSTRACT
Browsing concept lattices from Formal Concept Analysis becomes a problem as the number of concepts can grow significantly with the number of objects and attributes. Interpreting the lattice through direct graph-based visualisation of the Hasse diagram rapidly becomes difficult and more synthetic representations are needed. In this work we propose an approach to simplify concept lattices by extracting and visualising trees derived from them. We further simplify the browse-able trees with two reduction methods: fault-tolerance and concept clustering.

Keywords: Formal Concept Analysis, Tree, Visual Analytics, Reduction Methods.

INTRODUCTION
The vast amount of data generated over the last decades has brought new challenges to the analytics science. Visual data analysis and knowledge representation employ methods such as Formal Concept Analysis (FCA) in order to identify groupings of patterns from the analysis process (Ganter & Wille 1999). FCA provides an intuitive understanding of generalization and specialization relationships among data elements (objects) and their features (attributes or properties) in a structure known as a concept lattice. For this reason, FCA is extensively used in Information Retrieval Systems (Carpineto & Romano 1996, Carpineto & Romano 2005, Ducrou & Eklund 2006) to refine queries, restrict the search space, recommend related documents, and in particular, to browse document collections (Carpineto & Romano 2005). For example, in a collection of documents containing the word "Paris", some documents may also refer to "Restaurants", "Flight tickets" or "Concerts". In the concept represented by \{“Paris”,“Restaurants”\}, one may find documents about “Best view”, “Famous”, “Cheap”, and so forth. Each navigation step is a formal concept and the relationships expressed in the concept lattice allow navigating from one concept to the other.

Browsing concept lattices becomes a problem as the number of clusters grows significantly with the number of objects and attributes. Interpreting the lattice through a direct visualisation of the Hasse diagram rapidly becomes difficult and more synthetic representations are needed. A common approach is to show or hide parts of the lattice via interactive exploration of subsets of terms or neighbours of a focus concept (Ducrou & Eklund 2006). Carpineto et al.
(Carpineto & Romano 2005) defined constraints to be applied to the concept lattice in order to simplify lattice querying and navigation.

Trees are good alternatives to represent concept lattices in this context, because they do not suffer from edges crossings and they are natural metaphors for navigation history since there is only one parent per child. Additionally, users are familiar with that structure as trees are used to navigate through folders and files in most operating systems. In (Carpineto & Romano 2005) it was noted that trees are particularly interesting structures to represent concept lattices for browsing, however authors pointed that their main disadvantage is the amount of replicated information when concepts have multiple parents. In the current work we avoid duplication by selecting only one parent for each concept in the lattice. One inherent challenge is to formally define the notion of “best” parent among the potentially numerous parents of a concept. Previous work (Le Grand et al. 2009) used an approach to extract trees from lattices based on the assignment of weights to attributes. Concepts with higher average weights were selected as parents while the other concepts were removed from the resulting tree.

In this work we propose a tree-based approach for large lattice browsing, firstly by extracting and visualising trees derived from the lattice structure; Secondly, by enhancing the readability of the derived trees through colouring and distortion techniques. Finally, further simplification of the tree is done by using lattice reduction techniques. In order to extract trees from lattices, a single parent node needs to be chosen for each concept of the initial lattice: we therefore define a set of (unique) parent concept selection criteria, including the stability and support indexes (Kuznetsov 1990, Stumme et al. 2002) provided by FCA literature, as well as confidence and similarity measures. Before proceeding, we would like to briefly introduce FCA terminology and basic definitions used throughout the paper.

**A Brief Introduction to FCA**

Rudolf Wille and Bernhard Ganter introduced formal Concept Analysis in the 1990s based on the mathematisation of the philosophical notion of concepts and concepts hierarchy (Ganter & Wille 1999). A formal concept has an extent, composed of all objects that belong to the concept, and an intent, comprising all attributes (or properties) which apply to all objects of the extent. The set of objects and attributes, together with their relation to each other, form a formal context which is represented by a cross table, like the one illustrated in Table 1. Using the FCA terminology (Ganter & Wille 1999), given a (formal) context \( K = (G, M, I) \), where \( G \) is called a set of objects or extent, \( M \) is called a set of attributes or intent, and the binary relation \( I \subseteq G \times M \) specifies each object’s attributes. Derivation operators \( (\cdot)' \) are defined for \( A \subseteq G \) and \( B \subseteq M \) as:

\[
A' = \{ m \in M | \forall g \in A: gIm \}; \\
B' = \{ g \in G | \forall m \in B: gIm \}.
\]

A pair \( \{(A, B)\} \) where \( A \subseteq G \) (extent), \( B \subseteq M \) (intent) is a formal concept in a given formal context \( K \) only if \( A' = B \) and \( B' = A \). A formal concept is, therefore, a closed set of object/attribute relations, in that its extent contains all objects that have the attributes in its intent, and its intent contains all attributes shared by the objects of its extent. In the example illustrated in Table 1, it can be seen that \( \{\{“Flying”, “Bird”\}, \{“Finch”, “Eagle”\}\} \) is a formal concept because both birds are flying birds, and no other animal in this context is both a bird and a flying animal.
Table 1. A formal context of animals.

<table>
<thead>
<tr>
<th>Animal / Feature</th>
<th>Preying</th>
<th>Mammal</th>
<th>Flying</th>
<th>Bird</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lion</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Finch</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Eagle</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Hare</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Ostrich</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A partial ordering relation can be established among formal concepts, leveraging the so-called Galois connection. The partial ordering of concepts, or the concept lattice, is typically represented as a layered directed acyclic graph, named Hasse diagram, that illustrates the groupings of objects described by common attributes. Hasse diagrams such as the one shown in Figure 1 display the partially ordered sets (posets) between concepts in a hierarchical fashion, from the most generic (top) to the most specific concept (bottom). Each concept may have several parent concepts, as it depicts the various generalisation levels for each concept. The partial order among concepts of the lattice provides different levels of abstraction and native navigation links from a given concept.

![Hasse diagram](image)

Figure 1. Generated concept lattice for the animal’s context in Table 1.

This paper is organized as follows. Section 2 provides a literature review on lattice representations; Section 3 describes the overall approach and proposes a set of criteria for transforming concept lattices into trees; Section 4 discusses visualisation techniques to help users choose the parent selection criteria they are interested in. Section 5 presents a case study of the tree-extraction process in the tourism domain, followed by a Section dedicated to concept lattice reduction. Finally, Section 7 presents a discussion about the advantages and limitations of the proposed approach and Section 8 concludes the paper.
VISUAL REPRESENTATION OF CONCEPT LATTICES

The Hasse diagrams used in FCA are usually layered graphs, where concept vertices are assigned to horizontal layers according of the number of common attributes in each concept, and are ordered within each layer to reduce edge crossings. FCA lattices in particular suffer from considerable edge crossings, especially if the number of concepts exceeds a few dozen; this is unfortunately the case in most real word applications (Roth et al. 2006), which leads to reduced graph readability and aesthetics (Ware et al. 2002).

To reduce the complexity of lattices, simplified diagrams can be produced by displaying only concepts with a sufficient support (Stumme et al. 2002). Visualisations can also be restricted to portions of the data (Ducrou et al. 2006), and concept number reduction is possible by incorporating conditions into the data mining process (see Section 6). Finally, conceptual measures can be applied to identify the most significant concepts and filter outliers (Le Grand et al. 2009).

To deal specifically with the visual complexity of Hasse diagrams, several approaches allow users to dynamically explore and reveal specific parts of the diagram, using visual query languages (Cruz et al. 1987, Consens & Mendelzon 1993, Blau et al. 2002). However these techniques do not provide a clear view of the entire lattice.

Other FCA visualisation approaches map the distances between concepts to visual variables, in order to highlight patterns. For example in (Soto et al. 2009) similar concepts are represented as similarly coloured pixels placed in the 2D space along a Peano-Hilbert curve, so that similar concepts are displayed close from one another. Nevertheless, in these representations detailed relationships between concepts are lost. Finally, systems often provide users with hybrid/combined lattice visualisation, e.g. showing both a general Hasse diagram and a tag cloud for representing the neighbours of a specific concept (for a review see Eklund & Villerd 2010).

Our approach consists in representing lattices not as Hasse diagrams, but as trees. As we explain in the next section, we propose different criteria to extract trees from lattices and visualise the resulting trees. Trees are inherently simpler hierarchical structures than Hasse diagrams and due to their applicability in many domains, there is a plethora of tree representations. These include: indented outline trees, sometimes called “tree lists” (common in file browsers such as windows Explorer), traditional layered node-link diagrams in 2D or 3D (e.g. ConeTrees - Robertson et al. 1991), spatially transformed tree diagrams (e.g. Radial Eades 1992) as well as several space optimization (Space Optimized trees Nguyen & Huang 2002) and space-filling tree visualisation techniques (e.g. TreeMaps Johnson & Shneiderman 1991).

TREE-EXTRACTION FROM CONCEPT LATTICES

Trees are common and easily understandable visual representations. We consider them as a visualisation alternative to large cluttered concept lattices, which preserves all lattice entities and some of its structure. The tree-extraction process consists in choosing one single parent for each concept of the lattice. In order for a tree visualisation to be an effective alternative to a lattice, the extraction of the tree from the lattice needs to preserve the most essential features of the original structure. In the following sections we present two contrasting strategies for parent selection: the initial approach recursively selects one parent for all concepts in a given layer –
starting from the “leaves” of the lattice as explained in the following section. The second approach leverages conceptual measures to select the best parent candidates.

**Tree Extraction based on the Selection of One Parent per Layer**

In a previous work by Le Grand & Soto 2001, authors presented an approach to extract trees from concept lattices based on the selection of one parent concept per hierarchical layer in the lattice. The process starts by collecting all most specific concepts (i.e. the lower bound’s parent concepts at the second lowest layer of the concept lattice) so all objects are present in the new structure. A single parent concept is then selected for each of these concepts, and the process goes recursively until reaching the top of the lattice. The choice of a single parent for each concept therefore removes links and eventually concepts.

The goal is to select parents according to their relevance from the user’s point of view, for example, taking into account the attributes that are most significant to him or her. The challenge is to find the set of criteria that best suits users’ expectations. Trees extracted from the same concept lattice can be very different, as shown in Figures 2b and 2c, which represent two distinct trees extracted from the lattice of Figure 2a. Notice that concept C6 is lost in the tree transformation process of Figure 2c.

The trees are displayed in three different forms: the left one represents the tree as extracted from the lattice, the middle one is the traditional representation of a tree without crossing links, and finally the representation on the right represents the layers of the tree as fractions of concentric arcs, from the most generic (outer layers) to the most specific (inner layers); this visualisation is rather close from (Trad et al. 2009). The size of each arc is proportional to the number of children in the corresponding sub-tree.

**Figure 2a. Original concept lattice.**

---

1 If a concept is not selected as a single parent by any of its children concepts, it is removed from the lattice.
A hierarchy of parent selection criteria was defined in (Trad et al. 2010) to extract a tree from the concept lattice generated from on a corpus of 126 web pages dedicated to tourism. The algorithm works as follows:

- For each concept, called the current concept, it selects the parent concept which is located at the shortest distance from the upper bound of the lattice (in order to minimize the number of links);
- if several parent concepts satisfy this condition, it favours those which have already been selected as a single parent by other concepts of the same level as the current concept;
- If the last condition is not sufficient, it selects the concept with higher average of attribute weights, previously assigned by the user.

Figure 3 shows the conceptual tree extracted according to these criteria from the lattice generated from the tourism data set. Outer arcs represent most general concepts, i.e., those situated on the top in Hasse diagrams. To facilitate the interpretation of the tree, the intent of each of the concepts in the top layer is displayed. The surface of each arc is proportional to the number of leaves of the corresponding sub-tree to provide a glimpse of the most significant topics in the data set, in this case pages devoted to France, food, camping, and the region Loire.
Tree-extraction based on Conceptual Indexes

The initial tree extraction approach presented above has several drawbacks, the elimination of concepts which can lead to an undesired effect because of the loss of conceptual information. The present approach leverages conceptual measures as criteria to select parent concepts, keeping the link with the parent that scored the best with the considered index. As with the previous approach, the process starts with the most specific concepts i.e., at the bottom of the Hasse diagram and recursively computes the index for each of the candidates and eliminate links to all parent concepts except the one with the selected parent. The pseudo-code of the algorithm is provided in Table 2.

Choosing a single parent concept at each step leads to some information loss, although in this case only links are removed. Our goal is to minimize this loss by selecting parents using the most relevant criteria according to the kind of analysis performed by the analyst. In the following sections we consider various strategies for selecting parent concepts, including the stability (Kuznetsov 1990) and support (Stumme et al. 2002) indexes from FCA literature, as well as the confidence and similarity measures.
Algorithm ExtractTree

Concepts := \( \mathcal{B}(\mathcal{A}) \)

for each \((A, B)\) in Concepts
    Parents\([(A, B)] := \text{list of parent concepts of }(A, B)\)
    if \(|\text{Parents\([(A, B)]}\)| > 1
        max_score = -1
        for each \((C, D)\) in Parents\([(A, B)]\)
            Score\([(C, D)] := \text{the score for concept}(C, D)\)
            if Score\([(C, D)] > \text{max_score}
                selected_parents\([(A, B)] := (C, D)\)
                max_score := Score\([(C, D)]\)
            end if
        end for
    end if
end for
return selected_parents

Table 2. Algorithm to extract a tree from the concept lattice.

Parent Selection based on Stability and Support

The stability index measures the proportion of subsets of objects of a given concept whose derivation is equal to the intent of this concept (Kuznetsov 1990). In other words, stability indicates the probability of preserving a concept’s intent while removing some objects of its extent. A more stable concept is less dependent on individual members in the extension. We recall the definition of stability:

**Definition 1.** Let \( K = (G,M,I) \) be a formal context and \((A,B)\) be a formal concept of \( K \). The stability index of \((A,B)\) is defined as:

\[
\text{stability}(A,B) = \frac{|\{C \subseteq A | C' = B\}|}{2^{|A|}}
\]

Using the context in table 1 as an example, we calculate the stability for concepts 2 and 4 in order to select a parent for concept 5 (0.25 and 0.5 respectively); we keep the one with highest stability, in this case we therefore remove the edge between concepts 2 and 5. The idea behind the choice of the parent concept with the highest stability is that we expect to keep parent concept’s meaning even if some of the objects or attributes are removed. On the other hand, the support measure is the relation between the intent closure and the total number of objects (Stumme et al. 2002):

**Definition 2.** Let \( B \subseteq M \). The support count of the attribute set \( B \) in \( K \) is:

\[
\text{support}(B) = \frac{|B'|}{|G|}
\]
The use of support as parent selection criterion may lead to trees containing concepts with fewer specialization levels since generic concepts generally have higher support values than their most specific counterparts.

**Parent Selection Based on Similarity**

This measure relies on clustering parent and child concepts that share most of their attributes or objects. Parent and child having a great number of attributes in common are supposed to be grouped together following the principle of similarity clustering and local predictability (Hannan & Pogel 2006). It can be defined as:

**Definition 3.** Let ConceptParent \((A,B)\) be such that \(A \subset G\) and \(B \subset M\). Let ConceptChild \((C,D)\) be \(C \subset G\) and \(D \subset M\). The shared attribute index of an edge \(E (C,D) \rightarrow (A,B)\):

\[
\text{similarity}(E) = \frac{|B \cap D|}{|M|}
\]

In the same animal’s context of Table 1, we have potential parent concepts 2 and 4 sharing the same number of objects with concept 5, but concept 4 has more attributes in common with 5, so it should be chosen as the unique parent of concept 5 if the similarity criterion is considered.

**Parent Selection Based on Confidence**

The **confidence** value of a concept estimates how likely an object that has an attribute set \(A\), also has an attribute set \(C\) (Ganter & Wille 1999). In other words, it tries to measure how strong the implication of the parent concept to a child concept is. For instance, considering the formal context in Table 1, what is the probability of a given object that is \{Bird, Flying\} to be also \{Bird, Flying, Preying\}? The following paragraph formalizes its definition.

**Definition 4.** Let ConceptParent \((A,B)\) be such that \(A \subset G\) and \(B \subset M\). Let ConceptChild \((C,D)\) be \(C \subset G\) and \(D \subset M\). The confidence of an edge \(E (C,D) \rightarrow (A,B)\):

\[
\text{confidence}(E) = \frac{|C|}{|A|}
\]

An advantage of this method is its consistency with the interpretation of concept lattices. Taking our context as example (Figure 1), there is a 50% probability that an animal that is a flying bird is also a flying and preying bird. By contrast, a preying animal has only 33% of chance of being also a flying bird.

**VISUALISATION TECHNIQUES TO GUIDE USERS IN THE CHOICE OF THE PARENT SELECTION CRITERIA**

Common graph drawing techniques include the assignment of different colours, shapes and sizes to nodes and edges, according to different dimensions or properties. This approach is underused in traditional lattice visualisations, where the main visual variable used is node/link
colour to reflect user selections or node size to indicate the immediate presence of an extent or intent as displayed in ConExp\(^2\).

In our work we use these as well as other visual variables in a Hasse diagram to represent possible tree extraction criteria. For example, visualising different extraction criteria using various visual variables, allows users to compare these criteria in order to choose the one that better fits their needs. Prominent features of the lattice like specialization and generalization can be better understood: for instance, the power of implications of different concepts can be rendered by edge thickness. In this way users can be guided in understanding and choosing criteria for extracting trees to simplify the lattice representation.

We made some changes in the radial filling visualisation shown Figure 3. Because the number of concepts tends to increase in the most specific levels, the previous visualisation suffered from decreasing inner space. We decided to invert the way the hierarchy is displayed, top nodes are now in inner layers while more specific concepts are located in outer, more spacious layers. This visualisation layout is called “Sunburst” (Stasko et al. 2000) and provides an overview of the distribution (depicted as the size of the arcs) (Figure 4).

One challenge is to assign label to concepts in a tree visualisation like the sunburst. Concepts are labelled using elements from its intent and/or extent. It may become a real issue if the number of attributes or objects is too long to be displayed, causing text overlaps. In the sunburst for example, layers with a large number of concepts tend to have smaller arc size. A common technique to cope with this problem is to only assign labels that were not previously assigned to any of its parents, i.e., avoid repeating labels that already appeared in parent concepts. We used this technique combined with a logic that hides the label of a concept if the arc size is too small to fit the text. Optionally, the user can zoom in the areas of interest and hidden labels will appear accordingly.

Figure 4. Tree-extraction process and Sunburst visualisation.

CASE STUDY OF THE TREE-EXTRACTION PROCESS ON THE TOURISM WEB PAGES DATA SET

In this section we discuss a case study of a concept lattice to qualitatively examine the nature of the trees resulting from different criteria. Figure 5 shows the Hasse diagram of a lattice containing 2214 concepts and 7758 links, constructed from a corpus of 126 web pages dedicated to tourism, characterized by their most common words from 60 words extracted from thesaurus.

Figure 5. Concept lattice for the tourism web pages data set.

For the sake of the example, we selected a sub-context from the dataset containing only 87 concepts. In this particular context, the original concept lattice depicted by the Hasse diagram contained 216 edges between concepts, reduced to 82 after the tree extraction (38% of the original edges). Each of the proposed measures revealed particular aspects on the analysis of a lattice, as illustrated in Table 3.

The left column in Table 3 shows the complete tree extracted from the lattice using the corresponding criterion, whereas the column on the right displays a part of the tree used for discussion. As explained above, each concept is labelled in a non-repetitive way, i.e., labels assigned to parents are not be displayed in the respective children. Since labels can change from one visualisation to the other depending on the parent selected, colours were applied to each concept for easy identification (all concepts have the same colour in all visualisations e.g. loisirs/voyage/camping).

Table 3 a) shows the tree generated with extent support as parent selection criterion. This criterion tends to group concepts that have frequent objects in their extent. Notice that the concept containing {“montagne”} (mountain) accounted for many other concepts containing that attribute in contrast with confidence and similarity, for instance. This occurs because there are many web pages relating the keywords {“montagne”, “ski”} and {“montagne”, “restaurant”}, thus making the corresponding concepts more likely to be selected as parent in this criterion. On the other hand, the concept which contains {“ski”} has only one sub-concept {“voyage”} in contrast with the trees extracted using confidence and similarity (Table 3 c and d). We can infer that pages containing {“ski”} alone are not as popular as other keywords, because they are commonly associated with “montagne”, “transport” or {“loisir”, “voyage”} as the corresponding concepts show.
Table 3. Trees generated from the lattice in figure 4 for each proposed measure.

<table>
<thead>
<tr>
<th>Tree</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) Support (Extent)</td>
<td><img src="image1" alt="Tree Diagram" /></td>
</tr>
<tr>
<td>b) Stability (Intent)</td>
<td><img src="image2" alt="Tree Diagram" /></td>
</tr>
</tbody>
</table>
c) Similarity

d) Confidence
The stability (intent) index was the criterion that generated the tree in Table 3 b). With this index, concepts that retain the same set of web pages when some of their keywords are removed are the best candidates for parent selection. In the detail of Table 3 b) the concept containing \{“gastronomie”, “ski”, “montagne”\} was the preferred parent to append the child concept \{“gastronomie”, “ski”, “montagne”, “voyage”, “loisir”\} contrarily to the selection made in the tree using support (Table 3 a) where the parent is \{“montagne”, “ski”, “voyage”\}. This suggests that despite having fewer occurrences (web pages), the concept \{“gastronomie”, “ski”, “montagne”\} has a smaller set of keywords that together correspond to the same pages. For instance, pages that talk about “gastronomy in the mountains” can cite “ski” as one activity to do in the mountains but the absence of that keyword is not crucial to describe the page.

Table 3 c) depicts the tree generated by the similarity criterion. Parent concepts sharing most objects with a particular child concept were the best candidates. As an example, the concepts involving \{“gastronomie”, “ski”\} were appended to the concept \{“ski”\} because there are more pages in common with the later than there are pages shared with the concept \{“gastronomie”\}. A similar rationale explains the same parent selection for the confidence criteria: a page containing both keywords “gastronomie” and “ski” is more likely to be about skiing than about “gastronomie”. These choices contrast with those using stability and support (Table 3 a and b).

In Table 3 d) the tree was generated from confidence criterion, i.e. children concepts are associated with the parent to which the relationship of confidence is the highest among the candidates. As a consequence, the concept \{“voyage”\} was the parent selected for the concept containing \{“camping”, “voyage“\} as opposed to the concept \{“camping”\}, because usually camping implies going on a trip whereas going on a trip does not necessarily mean to go camping.

FURTHER SIMPLIFICATION THROUGH REDUCTION METHODS FOR FORMAL CONCEPT ANALYSIS

Although the browseable tree simplifies the concept lattice by pruning some of its edges, there may be still a number of irrelevant concepts in the remaining structure. This led us to investigate other reduction techniques for large concept lattices. In general, the reduction techniques for FCA are divided into two strategies: those that group similar objects or concepts, and those that act by removing irrelevant concepts. They can take place in the scaling phase, for instance, reducing the columns with low support or through the creation of sub-contexts (Andrews 2011), during the concept mining phase, e.g. keeping concepts with enough support (Stumme et al. 2002), or in the concept lattice, through visual cluttering reduction techniques (Soto et al. 2009). We used two known reduction techniques, fault-tolerance and clustering, from the FCA and data mining literature to simplify the navigation in tree derived from large concept lattices. We briefly describe both techniques in the following sections.

Fault-Tolerance

Reduction of formal contexts can be achieved by removing noise or outliers from the data. Noise refers to the data that does not fit in a grouping or concept when the difference is not important to a given purpose and it can drastically increase the number of concepts generated
(Boulicaut et al. 2003, Pensa & Boulicaut 2005, Andrews & Orphanides 2010). Noise is most common in real data such as surveys, logs, documents collections, when values can be biased, missing or when some mistake was introduced in the data. The notion of fault-tolerant FCA was introduced by (Pensa & Boulicaut 2005) to allow a certain number of ‘exceptions’ to occur in a concept. Based on the idea of “free-sets” (Boulicaut et al. 2003), the method seeks to find maximal rectangles of true values in the context bounded by δ exceptions (outliers). The resulting “purified” context is less likely to generate extra concepts to represent these exceptions in data and therefore it simplifies the concept lattice.

**Concept Clustering**

The notion of clustering is well known in the data-mining domain. Clustering of concepts (as opposed to clustering of objects or attributes) consists of grouping together concepts based on their affinities and can be useful to facilitate the browsing of concepts and to identify zones of interest. In our experiment, we used a spectral/k-means clustering algorithm (Hein et al. 2006) to identify clusters of concepts and reduce the number of browsing steps. To define the affinity matrix, we compute a Jaccard index with topological distance for each pair of concepts (A,B) and (C,D), A,C ∈ G and B,D ∈ M:

\[
sim((A,B), (C,D)) = \alpha \frac{A \cap C}{A \cup C} + (1 - \alpha) \times \frac{\text{dist}((A,B), (C,D))}{\text{maxDist}}
\]

Where \( \alpha \) is a ponderation factor [0,1] between the Jaccard index and the topological distance in the lattice (number of links). Dist is the shortest distance between two concepts in the lattice and maxDist is the diameter (maximum distance) in the lattice.

**DISCUSSION**

The extraction of a tree from the original lattice implies in a loss of information and risk of bias, depending on the selection criteria. Nevertheless, the process is able to keep some of the most essential conceptual features and provides different perspectives to the analysis: should it focus on frequent patterns or on the implications between the concepts? It is also possible to display different trees at once in order to get a more comprehensive view.

When two parent concepts have the same score, the parent with the first computed score is selected by the algorithm. This could lead to an undesired non-deterministic result for the same input parameters. We stress that even non-deterministically in this cases, the process yields results consistent with the chosen metric. One way of tackling this issue would be to define “tiebreak” scores, i.e., other indexes to be used to define the best parent. For instance, if two concepts have the same support, the parent selection second criteria can be stability.

The choice of parent selection criteria for tree transformation corresponds to a classification problem. Deciding if a Lion is more “mammal” than it is “preying” it’s not always straightforward, hence we rely on the measures that attempt to keep the context semantics when looking at the entire concept lattice. For instance, if we have more objects described by mammal which are “closer” to Lion than other concepts, then it may reasonable to be chosen as its parent. As general recommendations, one should use the criteria that best fits to their analysis task (Table 3).
Table 3. General guidelines on the usage of the proposed metrics

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
<th>Rationale</th>
<th>Suitable for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stability</td>
<td>It measures how likely a concept is to change if some of their attributes or objects are removed.</td>
<td>Stable concepts are less impacted by noise and usually represent strong correlation with real world entities (e.g.: a concept that encapsulates our notion of “mammal”).</td>
<td>Observing real world analogies</td>
</tr>
<tr>
<td>Support</td>
<td>It measures the frequency of the concept itemset.</td>
<td>Frequent concepts are usually generic concepts since they aggregate a larger number of objects than the specialized ones.</td>
<td>Frequent pattern analysis</td>
</tr>
<tr>
<td>Similarity</td>
<td>It represents the degree of similarity between parent and child nodes.</td>
<td>Concepts that share most attributes or objects should be linked together because they are similar.</td>
<td>Similarity analysis</td>
</tr>
<tr>
<td>Confidence</td>
<td>It measures how strong the implication is between a parent concept in a child concept.</td>
<td>Implication is one of the desired interpretation of a concept lattice.</td>
<td>Confidence analysis</td>
</tr>
</tbody>
</table>

CONCLUSIONS AND FUTURE WORK

In this work we have presented a transformation approach to extract trees from concept lattices, attempting to minimize both semantic and conceptual loss in favour of readability and interpretation. We have also presented ways to visually show the extraction criteria in the original lattice in order to help users choose the criteria adapted to their needs. This is an important step in the visual analysis of conceptual structures, as the resulting tree structures are visually easier to understand than cluttered lattice graphs. Domain experts can thus visually explore larger datasets that traditional visualisations of concept lattice cannot represent effectively. Each of the tree construction measures proposed in our work provides particular insights valuable to different analysis tasks, identified in our paper as recommendations.

One of the perspectives of this work is to define a quality measure to assess tree extracted from the conceptual lattice. Different criteria may be taken into account, for example:

- The depth of the tree, which indicates the number of levels of navigation that the user can explore;
- The balance of concepts distribution at each level;
- The proportion of original concepts lattice were eliminated.

In the future we plan to combine two or more criteria for parent selection with other lattice reduction techniques (e.g. icebergs lattices Stumme et al. 2002). We also plan to conduct user experiments to understand when users want to have full lattice views vs. tree views, which
metrics for creating trees are of most interest to them and under which circumstances, and assess whether our visual indications allow users to understand the tree extraction process.

ACKNOWLEDGEMENTS

This work is partly funded by the CUBIST project (“Combining and Uniting Business Intelligence with Semantic Technologies”), funded by the European Commission’s 7th Framework Programme of ICT, under topic 4.3: Intelligent Information Management.

REFERENCES


