Conceptual Knowledge Discovery with Frequent Concept Lattices

Gerd Stumme
Technische Universität Darmstadt, Fachbereich Mathematik,
Schloßgartenstr. 7, D–64289 Darmstadt; stumme@mathematik.tu-darmstadt.de

Abstract

Knowledge discovery support environments include beside classical data analysis tools also data mining tools. For supporting both kinds of tools, a unified knowledge representation is needed. We show that concept lattices which are used as knowledge representation in Conceptual Information Systems can also be used for structuring the results of mining association rules. Vice versa, we use ideas of association rules for reducing the complexity of the visualization of Conceptual Information Systems.

1 Introduction

The aim of *Knowledge Discovery in Databases* (*KDD*) is to support human analysts in the overall process of discovering useful information and knowledge in databases. Many real-world knowledge discovery tasks are both too complex to be accessible by simply applying a single learning or data mining algorithm and too knowledge-intensive to be performed without repeated participation of the domain expert. Therefore, knowledge discovery in databases is considered an interactive and iterative process between a human and a database that may strongly involve background knowledge of the analyzing domain expert. Following [Fayyad *et al.*, 1996], we understand KDD as the overall process of discovering useful knowledge from the data while *data mining* is considered as one step of KDD, namely the application of algorithms for extracting patterns from the data. In most applications, classical *data analysis* and *decision support* facilities (for instance Online Analytical Processing (OLAP) or statistical packages) are already present when data mining tools are added to the knowledge discovery support environment. For supporting the analyst in the overall process of human-centered knowledge discovery, both decision support and data mining tools should provide a homogeneous environment. In particular, this shows the need of a *unified knowledge representation.*

In this paper, we use *concept lattices* as such a unified knowledge representation for a knowledge discovery support environment which integrates *Conceptual Information Systems* and mining tools for *association rules.*

Association rules are statements of the type '37 % of the customers buying coffee also buy milk'. The task of mining association rules is to determine all rules that have a certain *confidence* (37 % in the example) and a certain *support* (the percentage of customers buying coffee and milk). Mining association rules can nowadays be considered as one of the core tasks of KDD.

Conceptual Information Systems are based on Formal Concept Analysis. *Formal Concept Analysis (FCA)* is a mathematical theory formalizing the concept of 'concept', introduced by Wille [1982]. During the years, FCA grew to a data analysis method [Ganter, Wille, 1999] which is now commercially applied by NaviCon Gesellschaft für Begriffliche Wissensverarbeitung mbH. In the past few years, FCA has been used by different AI researchers as a knowledge representation mechanism in various fields (e. g., [Schmitt, Saake, 1997], [Erdmann, 1998]). Stumme [1998] compares Conceptual Information Systems with OLAP and Stumme, Wille, and Wille [1998] discuss how FCA can support a human-centered knowledge discovery process called *Conceptual Knowledge Discovery in Databases (CKDD).*

Concept lattices are the knowledge representation of FCA. In Conceptual Information Systems, they are also used for visualizing the knowledge. We will show in this paper that concept lattices can also support the mining of association rules. The benefit of combining FCA and association rules is mutual:

1. Knowledge representation by concept lattices has to face the problem of exponential growth of the lattices. This is especially problematic when dealing with large data tables, for instance in the analysis of basket data for a supermarket. The management tool TOSCANA for Conceptual Information Systems [Vogt, Wille, 1994] solves this problem by vertically splitting the database and combining only those parts which are of interest for the actual query. In this paper, we present another approach (which can be combined with TOSCANA) borrowed from *association rules*: We prune horizontally all concepts with low support and keep only the *frequent concepts.*

2. Usually the algorithms for mining association rules return long lists of rules where many rules are not of interest to the market analyst. Different approaches have been made for reducing the list, for instance 'meta-mining' the list or defining the 'surprisingness' of rules. In this paper we show how the list of association rules can be structured and reduced by using frequent concepts.

In the next section, we present the basics of FCA and association rules as far as they are needed for this paper. For more detailed introductions, refer for instance to [Ganter, Wille, 1999] and [Agrawal *et al.*, 1996]. In Section 3, frequent concepts are introduced, and in Section 4, we discuss how they can help structuring and reducing the mining of association rules.

2 Basics of Formal Concept Analysis and Association Rules

2.1 Formal Concept Analysis

Since concepts are necessary for expressing human knowledge the knowledge discovering process benefits from a comprehensive formalization of concepts. FCA offers such a formalization by mathematizing concepts that are understood as units of thought constituted by their extension and intension. This understanding of 'concept' is first mentioned explicitly in the Logic of Port Royal [Arnaud, Nicole, 1668] and has been established in the German standards DIN 2330 and DIN 2331.

To allow a mathematical description of extensions and intensions, FCA starts with a *(formal) context* defined as a triple $\mathbb{K} := (G, M, I)$, where G is a set of *objects*, M is a set of *attributes*, and I is a binary relation between G and M (i.e. $I \subseteq G \times M$). $(g, m) \in I$ is read "the object g *has* the attribute m".

Figure 1 shows the formal context $\mathbb{K}_{\text{coffee}} := (G_{\text{coffee}}, M_{\text{coffee}}, I_{\text{coffee}})$ where the object set G_{coffee} comprises all coffees sold by a supermarket and the attribute set M_{coffee} provides some attributes describing them.

	Jacobs	Plus	classic	mild	light	< 6 DM	< 8 DM	> 8 DM
Dallmayr Prodomo			×					×
Jacobs Krönung	×		×				×	
Jacobs Krönung Light	×				×		×	
Jacobs Krönung Free	×				×		×	
Jacobs Krönung Mild	×			×			×	
Jacobs Meisterröstung	×		×				×	
Tempelmann			×					×
Plus Schonkaffee		×			×		×	
Plus Naturmild		×		×		×	×	
Plus milde Sorte		×		×		×	×	
Plus Gold		×	×			×	×	
Idee Kaffee Classic			×					×
Kaffee Hag klassisch			×					×
Melitta Cafe Auslese			×				×	
Melitta Cafe Auslese Mild				×			×	
Kaisers Kaffee Auslese Mild				×			×	

Figure 1: The formal context $\mathbb{K}_{\text{coffee}}$

For $A \subseteq G$, we define $A' := \{m \in M \mid \forall g \in A\colon (g, m) \in I\}$ and, for $B \subseteq M$, we define $B' := \{g \in G \mid \forall m \in B\colon (g, m) \in I\}$. (In Sections 3 and 4, we will use the fact that $B \subseteq B''$, $B' = B'''$, and $(B_1 \cup B_2)' = B_1' \cap B_2'$ for all $B \subseteq M$. The same holds for $B \subseteq G$.)

A *formal concept* of a formal context (G, M, I) is defined as a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$. The sets A and B are called the *extent* and the *intent* of the formal concept (A, B). The *subconcept–superconcept relation* is formalized by $(A_1, B_1) \leq (A_2, B_2) :\Longleftrightarrow A_1 \subseteq A_2 \quad (\Longleftrightarrow B_1 \supseteq B_2)$. The set of all concepts of a context $\mathbb{K}$ together with the order relation $\leq$ is always a complete lattice,[1] called the *concept lattice* of $\mathbb{K}$ and denoted by $\underline{\mathfrak{B}}(\mathbb{K})$. Figure 2 shows the concept lattice of the context in Figure 1 by a line diagram.

In the *line diagram*, the name of an object g is always attached to the circle representing the smallest concept with g in its extent; dually, the name of an attribute m is always attached to the circle representing the largest concept with m in its intent. This allows us to read the context relation from the diagram because an object g has an attribute m if and only if there is an ascending path from the circle labeled by g to the circle labeled by m. The extent of a concept consists of all objects whose labels are below in the diagram, and the intent consists of all attributes attached to concepts above in the hierarchy. For example, the concept labeled by '< 6 DM' has {'Plus Naturmild', 'Plus milde Sorte', 'Plus Gold'} as extent, and {'< 6 DM', 'Plus' (the house brand of the supermarket), '< 8 DM'} as intent.

For $X, Y \subseteq M$, we say that the *implication* $X \to Y$ *holds* in the context, if each object having all attributes in X also has all attributes in Y. For instance, the implication $\{\text{Plus, classic}\} \to \{< 6\,\text{DM}\}$ holds in the coffee context. It can be read directly in the line diagram: the largest concept having both 'Plus' and 'classic' in its intent is below the concept labeled by '< 6 DM'.

A *Conceptual Information System* consists of a *many-valued context* and a set of *conceptual scales*. A many-valued context may not only have crosses (i.e., yes/no) as entries, but attribute-value pairs. More precisely, a *many-valued context* is a tuple $\mathbb{K} := (G, M, (W_m)_{m \in M}, I)$ where G is a set of objects, M a set of attributes, W_m the set of possible values for the attribute $m \in M$, and the relation $I \subseteq G \times \{(m, w) \mid m \in M, w \in W_m\}$ [with $(g, m, w_1) \in I, (g, m, w_2) \in I \Rightarrow w_1 = w_2$] indicates if an object $g \in G$ has value $w \in W_m$ for attribute $m \in M$. A *conceptual scale* for a subset $B \subseteq M$ of attributes is a (one-valued) formal context $\mathbb{S}_B := (G_B, M_B, I_B)$ with $G_B \subseteq \times_{m \in B} W_m$. (The idea is to replace the attribute values in W_m which are often too specific by more general attributes which are provided in M_B. For an example, see below.)

For a basket data analysis of a supermarket, we consider as set G_{trx} of a many-valued context $\mathbb{K}_{\text{trx}}$ the set of all *transactions* of the supermarket (more precisely, their IDs); and as set M_{trx} of attributes the set of all

[1] I. e., for each subset of concepts, there is always a greatest common subconcept and a least common superconcept.

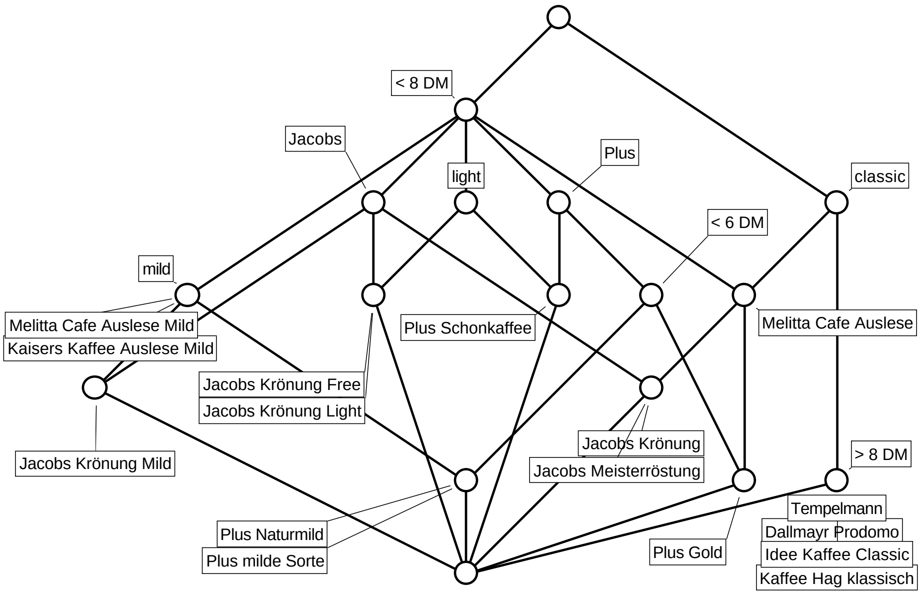

Figure 2: The concept lattice $\underline{\mathfrak{B}}(\mathbb{K}_{\text{coffee}})$ of the context $\mathbb{K}_{\text{coffee}}$

items sold by the supermarket together with the two attributes `time` and `date`. (One could add other attributes like `credit card number` etc.) For all items of the supermarket the attribute set is Boolean, e. g., $W_{\text{Dallmayr Prodomo}} = \{\text{yes}, \text{no}\}$. For an item $m \in M_{\text{trx}}$, we let $(g, m, \text{yes}) \in I_{\text{trx}}$ if item m was purchased (at least once) in transaction g, and $(g, m, \text{no}) \in I_{\text{trx}}$ else. For `time` we have W_{time} =[9.00, 19.59], and W_{date} contains all dates during the period to be analyzed.

An example of a conceptual scale for B={`time`} is given in Figure 3. The attributes M_{time}={morning, ..., evening} have been chosen because the analyst is usually not interested in seeing the exact time of the transactions, but is thinking in more general terms. When the analyst chooses the scale $\mathbb{S}_{\text{time}}$ in the management system TOSCANA, then the diagram in Figure 3 is displayed, but instead of the WHERE-clauses of SQL queries (e. g., time in [12.00, 13.59]) the numbers of transactions which fulfill the queries are shown. (This is the derived context for $\mathcal{S} = \{\mathbb{S}_{\text{time}}\}$ as defined below.)

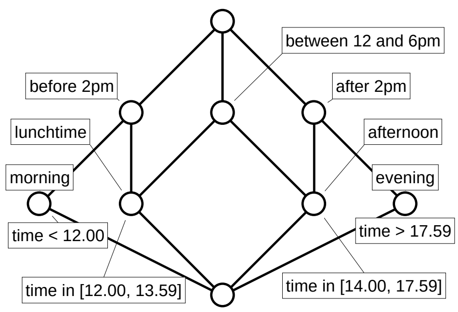

Figure 3: The conceptual scale $\mathbb{S}_{\text{time}}$

For B = {Dallmayr Prodomo, ..., Kaisers Kaffee Auslese Mild}, the formal context $\mathbb{K}_{\text{coffee}}$ cannot be used directly as a conceptual scale. The context we need must have the powerset $\mathfrak{P}(G_{\text{coffee}})$ as set of objects, not the set G_{coffee} itself, because customers can buy arbitrary combinations of coffee. This is provided by the conceptual scale $\mathbb{S}_{\text{coffee}} := (\mathfrak{P}(G_{\text{coffee}}), M_{\text{coffee}}, J_{\text{coffee}})$ with $(A, m) \in J_{\text{coffee}} :\iff \exists g \in A \colon (g, m) \in I_{\text{coffee}}$.

Now let $\mathfrak{S}$ be the set of conceptual scales for the many-valued context $\mathbb{K} := (G, M, (W_m)_{m \in M}, I)$. For any subset $\mathcal{S} \subseteq \mathfrak{S}$ of scales, we can now translate the many-valued context into a one-valued one: The *derived context* $\mathbb{K}_{\mathcal{S}}$ is defined by $\mathbb{K}_{\mathcal{S}} := \left(G, \bigcup_{\mathbb{S}_B \in \mathcal{S}} M_B, I_{\mathcal{S}}\right)$ with $(g, n) \in I_{\mathcal{S}}$ if there is a scale $\mathbb{S}_B \in \mathcal{S}$ with $m \in M_B$ and $w \in W_m$ with $(g, m, w) \in I$ and $(g, n) \in I_B$. For instance, if $\mathcal{S} = \{\mathbb{S}_{\text{time}}\}$, and (TID 0815, `time`, 11.17am)$\in I$, then we have (TID 0815, morning), (TID 0815, before 2pm)$\in I_{\mathcal{S}}$ [because (time < 12.00, morning)$\in I_{\text{time}}$ and (time < 12.00, before 2pm)$\in I_{\text{time}}$ as one can see in Figure 3].

One can hence derive from each many-valued context $\mathbb{K}$ one large one-valued context $\mathbb{K}_{\mathfrak{S}}$, for which the concept lattice could be computed. However, this approach is not feasible because the resulting lattice is usually much too large, and nobody wants to see it as a whole. Instead, the system TOSCANA allows to combine the diagrams of two (or more) conceptual scales in a *nested line diagram*: In each concept of the first scale, the second scale is inserted.[2]

[2] This works well because the lattice of the whole derived context can always be embedded (as a join-semilattice) in the direct product of all the scales. Because of space limitation,

For instance, by combining $\mathbb{S}_{\text{time}}$ and $\mathbb{S}_{\text{coffee}}$, the analyst can see how the types of coffee purchased change during the day: Is classical coffee bought more often in the morning, and light coffee in the evening? Hence if the analyst already guesses that there is some relationship between the time of the purchase and the type of coffee purchased, then the Conceptual Information System supports him in analyzing the situation in more detail. Supporting the user in finding such hypotheses is the task of mining association rules.

2.2 Association Rules

We can describe association rules in terms of Formal Concept Analysis: Consider again the context $\mathbb{K}_{\text{trx}}$. For the moment, we restrict the set M of attributes to the items sold by the supermarket (and ignore the many-valued attributes `time` and `date`). Then we can see the restricted context $\mathbb{K}^{\circ}_{\text{trx}}$ as a one-valued context. Each subset X of M is called an *itemset*. The *support* of X is defined by $\text{supp}(X) := \frac{|X'|}{|G|}$ (where $|G|$ is the cardinality of G).

An *association rule* $X \to Y$ consists of two subsets X and Y of M. We say that the rule $X \to Y$ holds with *support* $\text{supp}(X \to Y) := \frac{|(X \cup Y)'|}{|G|}$ and with *confidence* $\text{conf}(X \to Y) := \frac{\text{supp}(X\cup Y)}{\text{supp}(X)}$ (in short: $X \xrightarrow{s,c} Y$ with $s := \text{supp}(X \to Y)$ and $c := \text{conf}(X \to Y)$). (An implication is hence an association rule with confidence 1 and arbitrary support.)

Rules that hold only with a certain confidence have been investigated for a long time by many researchers. For instance, in the framework of FCA, Luxenburger [1991] has called them *partial implications*. The notion of association rules (which additionally have high support) and their application to large databases was introduced by [Agrawal *et al.*, 1993]. They stated the following problem and provided a first algorithm: Compute, for given $s_{\min}, c_{\min} \in [0,1]$, all association rules $X \xrightarrow{s,c} Y$ with $s \geq s_{\min}$ and $c \geq c_{\min}$.

There are now several algorithms for mining association rules in the literature. All algorithms work in two steps. First they determine the set $\mathcal{F}$ of all *frequent itemsets*, i.e., $\mathcal{F} := \{Y \in M \mid \text{supp}(Y) \geq s_{\min}\}$. Then they determine, for each $Y \in \mathcal{F}$, all $X \subseteq Y$ with $\text{conf}(X \to Y) \geq c_{\min}$. The expensive step is the first one. Hence almost all research effort is focussed on that step. In this paper, we focus on structural aspects of association rules, and discuss algorithms only briefly.

Generalized association rules have been introduced in [Srikant, Agrawal, 1995] because the association rules obtained by mining directly the large context $\mathbb{K}^{\circ}_{\text{trx}}$ with all items as attributes returns rules which are often too specific, for instance '37% of customers buying Jacobs Meisterröstung also buy Bärenmarke Kaffeemilch 0.25*l*', instead of '39% of customers buying classically roasted coffee also buy coffee milk'.

we refer to [Vogt, Wille, 1994] for an example.

For generalized association rules, one considers additionally a taxonomy on the set M°_{trx} of items. The taxonomy is a partially ordered set $(\mathcal{T}, \leq)$ (in which usually the items (i.e., the elements in M°_{trx}) are considered as the minimal elements). All other elements are called *generalized items*. We say that *transaction* $g \in G^{\circ}_{trx}$ *contains the generalized item* $t \in \mathcal{T}$ if there is a (non-generalized) item $m \in M^{\circ}_{\text{trx}}$ with $(g, m) \in I$ and $m \leq t$ in $(\mathcal{T}, \leq)$. For instance, if 'Plus Schonkaffee' $\leq$ 'light coffee' in the taxonomy, then each transaction containing 'Plus Schonkaffee' also contains the generalized item 'light coffee'. For mining generalized association rules, one could first add all generalized items to the context, and then mine that as a flat table. But this approach is quite inefficient, and all existing algorithms try to use the taxonomy to support pruning. Weber [1998] gives an overview over algorithms for mining association rules and generalized association rules.

Up to now, we have stated the basics of both Formal Concept Analysis and association rules. Now let us see how both theories can enrich each other.

3 Frequent Concept Lattices

Concept lattices provide exactly the same information than the formal context they are derived from. While this is a big advantage over other data analysis techniques in many applications, it is a serious handicap for large datasets. As mentioned before, this problem is usually faced by vertically splitting the database by using conceptual scales and displaying only a part by combining two or more scales in nested line diagrams. Here we consider a horizontal pruning of the concept lattice. As we will see later, both approaches can be combined.

For a given $s_{\min} \in [0,1]$, we define the *frequent concepts* of $\mathbb{K} := (G, M, I)$ as the concepts $(A, B) \in \underline{\mathfrak{B}}(\mathbb{K})$ with $\frac{|A|}{|G|} \geq s_{\min}$. The lattice $\underline{\mathfrak{B}}_{s_{\min}}(\mathbb{K}) := \{(A,B) \in \underline{\mathfrak{B}}(\mathbb{K}) \mid \frac{|A|}{|G|} \geq s_{\min}\} \cup \{(M', M)\}$ is called the *frequent concept lattice* of the context $\mathbb{K}$.[3]

By fixing a suitable threshold $s_{\min}$, we can now considerably reduce the concept lattice $\underline{\mathfrak{B}}(\mathbb{K}^{\circ}_{\text{trx}})$ to the frequent concept lattice $\underline{\mathfrak{B}}_{s_{\min}}(\mathbb{K}^{\circ}_{\text{trx}})$. The latter contains still all relevant information for the basket data analysis.[4] The frequent concept lattice is usually still too large to be displayed as a whole. But now, we can combine this horizontal pruning of the lattice with the vertical splitting of the data table: For each *conceptual scale* $\mathbb{S}_B$ (i.e., each 'slice' of the context $\mathbb{K}^{\circ}_{\text{trx}}$) we only display its frequent concept lattice $\underline{\mathfrak{B}}_{s_{\min}}(\mathbb{S}_B)$. The frequent concept lattice $\underline{\mathfrak{B}}_{s_{\min}}(\mathbb{K})$ of the total context $\mathbb{K}$ can then be embedded

[3] We have to add the smallest concept of $\underline{\mathfrak{B}}(\mathbb{K})$, (M', M), in order to obtain a lattice again. This is more a technical detail; see Footnote 5.

[4] Observe that the restriction to the frequent concepts is not suitable for other kinds of applications. For instance, in Conceptual Information Systems used for Information Retrieval, one is especially interested in the concepts with low support.

(as a join-semilattice) in the direct product of the frequent concept lattices of the conceptual scales (compare with Footnote 2). Hence one can still use the visualization method by nested line diagrams as it is implemented in TOSCANA. The use of frequent concept lattices allows us to work with conceptual scales which are too large to be displayed completely. For instance, the conceptual scale $\mathbb{S}_{\text{coffee}}$ which we introduced in Section 2 has 99 concepts. But it is only so large in order to cover all eventualities: Each of the $2^{16} = 65536$ combinations of coffees is considered in the scale. But with a reasonable threshold $s_{\min}$ for the support, we can assume that only single coffees and very few combinations of two different coffees are bought together frequently. Then the resulting lattice is not much larger than the concept lattice in Figure 2 and can be combined with another scale (for instance $\mathbb{S}_{\text{time}}$) in a nested line diagram. If there are no frequent combinations of two different coffees, then the lattices are even identical.

For computing the frequent concept lattice of a context $\mathbb{K}$, one can apply the Next-Closure-Algorithm (1984) of B. Ganter in [Ganter, Wille, 1999]. It is usually used for computing concept lattices, but can be used for determining arbitrary closure systems. A *closure system* $\mathcal{C} \subseteq \mathfrak{P}(M)$ on a set M is a set of subsets of M such that for any subset $\mathcal{X} \subseteq \mathcal{C}$, $\bigcap \mathcal{X}$ is a *closure* again, i.e., $\bigcap \mathcal{X} \in \mathcal{C}$.[5] To each closure system is assigned a *closure operator* $\overline{}\colon \mathfrak{P}(M) \to \mathfrak{P}(M)$ which maps each subset X of M to the smallest closure $\overline{X}$ containing X.

We briefly recall the Next-Closure-Algorithm. For a given closure operator, it determines all closures in the *lectic order*. For simplicity, we assume that $M = \{1,\ldots,n\}$. For $X, Y \subseteq M$, we say that $X <_i Y$ if and only if $A \cap \{1,\ldots,i-1\} = B \cap \{1,\ldots,i-1\}$ and $i \in B \setminus A$. Then the *lectic order* is defined by $X < Y \iff \exists i \in M\colon X <_i Y$. The lectic order is a total order on $\mathfrak{P}(M)$, i.e., for $X, Y \in \mathfrak{P}(M)$, we have always $X < Y$ or $X = Y$ or $X > Y$.

Algorithm: The lectically smallest closure is $\overline{\emptyset}$. For a given set $X \in M$, the lectically next closure is determined by: 1. Let $i := n$.

2. While $A \not<_i \overline{(A \cap \{1,\ldots,i-1\}) \cup \{i\}}$, do $i := i - 1$.
3. Then $\overline{(A \cap \{1,\ldots,i-1\}) \cup \{i\}}$ is the lectically next closure. The last closure is M. □

The intents of a concept lattice form a closure system, and can hence be determined by the algorithm with the closure operator $\overline{X} := X''$. For determining the frequent concept lattice $\underline{\mathfrak{B}}_{s_{\min}}(\mathbb{K})$, we have to modify the closure operator: $\overline{X} := X''$ if $\operatorname{supp}(X) \geq s_{\min}$ and $\overline{X} := M$ else. Since $X \subset Y$ implies $X < Y$, the algorithm prunes then all itemsets which have an infrequent itemset as proper subset.[6]

[5]Remark that $M = \bigcap \emptyset$ is always a closure. That is the reason why we had to add (M', M) to $\underline{\mathfrak{B}}_{s_{\min}}(\mathbb{K})$.

[6]Prutax [Hipp *et al.*, 1998], a depth-first algorithm for mining generalized association rules, traverses the power set $\mathfrak{P}(M)$ in the lectic order, too.

In the next section, we discuss how frequent concepts can be used for structuring and reducing the results of mining association rules.

4 Structuring Association Rules

In this section we show that it is not necessary to know all frequent itemsets for computing the relevant association rules. It is sufficient to consider intents of frequent concepts.

Let us call the intent of a frequent concept *frequent intent.* I.e., $X \subseteq M$ is a frequent intent if and only if $X = X''$ and $\operatorname{supp}(X) \geq s_{\min}$. We will see that instead of providing *all* association rules to the market analyst, we can restrict ourself to those rules $X \xrightarrow{s,c} Y$ where X and Y are frequent intents, together with a set of implications, called *frequent Duquenne-Guigues-basis*, which describes the structure of the frequent concept lattice $\underline{\mathfrak{B}}_{s_{\min}}(\mathbb{K})$.

The intents of a given context $\mathbb{K} := (G, M, I)$ are exactly those subsets of M which are closed under all implications which hold in $\mathbb{K}$. Hence it is sufficient to know how to generate all implications that hold in $\mathbb{K}$. A *basis* of implications is a set of implications from which one can derive all implications by using the following three rules [Amstrong, 1974]: *(1)* $X \to X$ for all $X \subseteq M$. *(2)* If $X \to Y$ then $X \cup Z \to X$ for any $Z \subseteq M$. *(3)* If $X \to Y$ and $Y \cup Z \to W$, then $X \cup Z \to W$.

Duquenne and Guigues [1986] have shown that the set of all implications $P \to P''$ where P is a pseudo-intent forms a minimal basis. A *pseudo-intent* is a subset P of M with $P \neq P''$ such that, for each pseudointent $Q \subseteq P$ with $Q \neq P$, $Q'' \subseteq P$ holds.

As we are interested in describing the frequent concept lattice only, we can prune the Duquenne-Guigues-Basis: We define the *frequent Duquenne-Guigues-basis* as the set $\{P \to P'' \mid P \text{ pseudo-intent}, \operatorname{supp}(P) \geq s_{\min}\}$. This set generates now all frequent implications, i.e., all association rules with high support and confidence 1.

The following theorem shows that for determining the remaining association rules (those with confidence $\neq 1$), we can restrict ourselves to those rules where both premise and conclusion are frequent intents. The proof is straightforward. For the confidence, it goes back to [Luxenburger, 1991].

Theorem. Let $X, Y \subseteq M$. Then $X \to Y$ and $X'' \to Y''$ have the same support and the same confidence.

We can now present the results to the market analyst in two parts: We provide the frequent Duquenne-Guigues-Basis together with the list of all association rules $X \xrightarrow{s,c} Y$ with $X = X''$, $Y = Y''$, $s \geq s_{\min}$ and $c \geq c_{\min}$. From these two lists, we can check whether an association rule $X \to Y$ holds with support $s \geq s_{\min}$ and confidence $c \geq c_{\min}$ in two steps: First we determine the implication $X \to X''$ by applying the implications from the frequent Duquenne-Guigues-Basis to the set X. Similarly we determine Y''. Then we can check whether $X'' \xrightarrow{s,c} Y''$ is provided in the second list.

By using these two lists, we can save the user from reading redundant association rules. The gain of our approach depends on how many frequent itemsets are *not* frequent intents. While itemsets with very few items tend to be intents (because there are transactions which have exactly these items in common), the more items an itemset has (and the lower its support is), the higher is the chance that the itemset is not an intent.

The gain is higher when we deal with generalized association rules. For instance, the implication $\{> 8\,\text{DM}\} \rightarrow \{\text{classic}\}$ will hold in any case; and it is not unlikely that the implication $\{\text{Plus, classic}\} \rightarrow \{< 6\,\text{DM}\}$ will hold as well.

5 Outlook

We have shown in this paper that bringing together Formal Concept Analysis and association rules can enrich both theories. Not all questions are solved yet, and further research is needed. We briefly state three interesting questions:

1. Implications can be read directly from the line diagram, which is more accepted by the users than a long list of implications. Due to the fact that association rules are not transitive, their visualization is much more difficult. The modification of line diagrams such that they also visualize association rules is one topic of further research.

2. The Next-Closure-Algorithm is not optimized for contexts with $|G| \gg |M|$, the typical situation in supermarket basket data analysis. Further research is needed to adapt existing data mining tools (which are optimized for this situation) such that they can compute the frequent pseudointents. (The computation of the frequent intents can easily be integrated in the existing algorithms, since $X \subseteq M$ is an intent if and only if there is no $m \in M \setminus X$ with $\text{supp}(X \cup \{m\}) = \text{supp}(M)$.)

3. A promising approach is to consider Conceptual Information Systems as preprocessing tools for mining association rules. Conceptual scales can be used as taxonomies for generalized association rules; and by selecting scales one can restrict the data to be mined and the level of detail on which the mining shall take place.

References

[Agrawal *et al.*, 1993] R. Agrawal, T. Imielinski, A. Swami: Mining association rules between sets of items in large databases. *Proc. ACM SIGMOD*, 1993

[Agrawal *et al.*, 1996] R. Agrawal, H. Mannila, R. Shrikant, H. Toivonen, A.I. Verkamo: Fast discovery of association rules. In [Fayyad *et al.*, 1996], 307–328

[Amstrong, 1974] W.W. Amstrong: Dependency structures of data base relationships. *IFIP Congress*, Geneva, Switzerland, 1974, 580–583

[Arnaud, Nicole, 1668] A. Arnauld, P. Nicole: La logique ou l'art de penser — contenant, outre les règles communes, plusieurs observations nouvelles, propres à former le jugement. Ch. Saveux, Paris 1668

[Erdmann, 1998] M. Erdmann: Formal Concept Analysis to Learn from the Sisyphus-III Material. *Proc. KAW '98*, Banff, Canada, 1998

[Fayyad *et al.*, 1996] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (eds.): *Advances in Knowledge Discovery and Data Mining.* AAAI/MIT Press, Cambridge 1996.

[Ganter, Wille, 1999] B. Ganter, R. Wille: *Formal Concept Analysis: Mathematical Foundations.* Springer, Berlin-Heidelberg 1999

[Guigues, Duquenne, 1986] J.-L. Guigues, V. Duquenne: Familles minimales d' implications informatives resultant d' un tableau de données binaires. *Math. Sci. Humaines* **95**, 1986, 5–18

[Hipp *et al.*, 1998] J. Hipp, A. Myka, R. Wirth, U. Güntzer: A new algorithm for faster mining of generalized association rules. LNAI **1510**, Springer, Heidelberg 1998

[Luxenburger, 1991] M. Luxenburger: Implications partielles dans un contexte. *Mathématiques, informatique et sciences humaines* **113**, 1991, 35–55

[Schmitt, Saake, 1997] I. Schmitt, G. Saake: *Merging Inheritance Hierarchies for Schema Integration based on Concept Lattices.* Preprint 2, Fakultät für Informatik, Universität Magdeburg 1997

[Srikant, Agrawal, 1995] R. Srikant, R. Agrawal: Mining Generalized Association Rules, Proc. VLDB 1995

[Stumme, 1996] G. Stumme: Local Scaling in Conceptual Data Systems. LNAI **1115**, Springer, Heidelberg 1996, 308–320

[Stumme, 1998] G. Stumme: On-Line Analytical Processing with Conceptual Information Systems. *Proc. 5th Intl. Conf. of Foundations of Data Organization.* Kobe, November 12-14, 1998

[Stumme, Wille, Wille, 1998] G. Stumme, R. Wille, U. Wille: Conceptual Knowledge Discovery in Databases Using Formal Concept Analysis Methods. LNAI **1510**, Springer, Heidelberg 1998, 450–458

[Vogt, Wille, 1994] F. Vogt, R. Wille: TOSCANA – A graphical tool for analyzing and exploring data. LNCS **894**, Springer, Heidelberg 1995, 226–233

[Weber, 1998] I. Weber: On pruning stragies for discovery of generalized and quantitative association rules. *Proc. Workshop on Knowledge Discovery and Data Mining, PRICAI.* Singapore, Nov. 22–27, 1998

[Wille, 1982] R. Wille: Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. In I. Rival (ed.): *Ordered Sets.* Boston-Dordrecht: Reidel, 1982, 445–470.