A smooth introduction to symbolic methods for knowledge discovery

Amedeo Napoli

A Smooth Introduction to Symbolic Methods for Knowledge Discovery

Amedeo Napoli
LORIA – UMR 7503
BP 239, F-54506 Vandœuvre-lès-Nancy
Amedeo.Napoli@loria.fr

Abstract

In this research report, we present a smooth introduction to symbolic methods for knowledge discovery in databases (KDD). The KDD process is aimed at extracting from large databases information units that can be interpreted as knowledge units to be reused. This process is based on three major steps: the selection and preparation of data, the data mining operation, and finally the interpretation of the extracted units. The process may take advantage of domain knowledge embedded in domain ontologies, that may be used at every step of the KDD process. In the following, we detail three symbolic methods for KDD, i.e. lattice-based classification, frequent itemset search and association rule extraction. Then, we present three applications of the KDD process, and we end this research report with a discussion on the the main characteristics of the KDD process.

1 Introduction

Knowledge discovery in databases can be likened to the process of searching for gold in the rivers: the gold nuggets that are researched are knowledge units, and the rivers are the databases under study. Huge volumes of data –and particularly documents– are available, without any intended usage. A fundamental question is to know if there may be something interesting in these data, and to find methods for extracting these "interesting things". The knowledge discovery in databases process –hereafter KDD– consists in processing a huge volume of data in order to extract knowledge units that are non trivial, potentially useful, significant, and reusable. Generally, the KDD process is iterative and interactive, and controlled by an expert of the data domain, called the *analyst*, who is in charge of guiding the extraction process, on the base of his objectives, and of his domain knowledge. The analyst selects and interprets a subset of the units for building "models" that will be further considered as knowledge units with a certain plausibility. The KDD process is based on three major steps: (i) the data sources are prepared to be processed, (ii) then they are mined, and (iii)

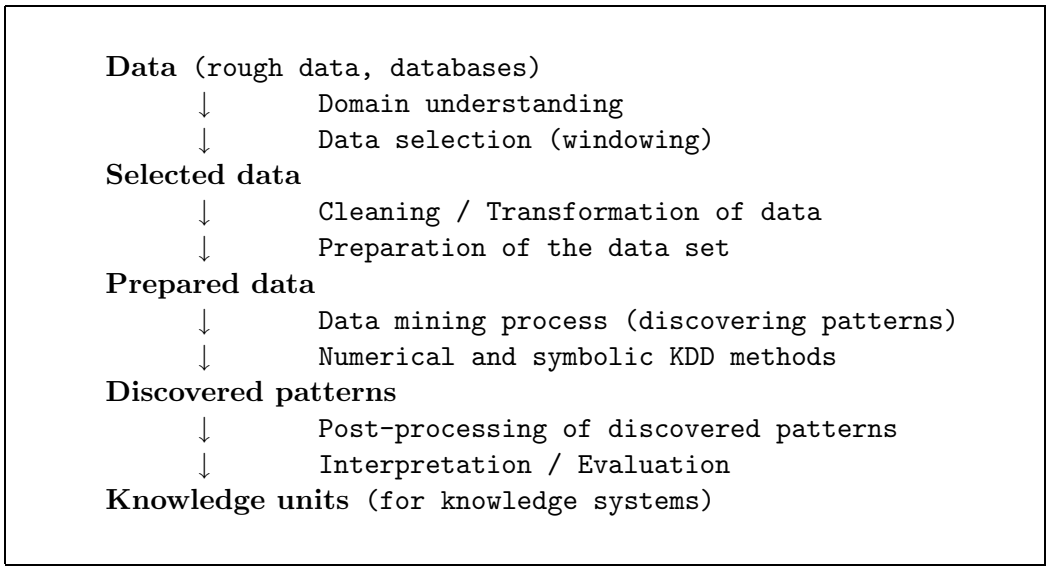

Figure 1: The KDD loop: from rough data to knowledge units. The overall objective process of the KDD process is to select, prepare and extract knowledge units from different sources, and then to represent the extracted knowledge units in adequate knowledge structures.

finally, the extracted information units are interpreted for becoming knowledge units. These units are in turn embedded within a representation formalism to be used within a knowledge-based system. The KDD process may also be understood as a process turning data into information and then knowledge (see figure 1), considering the following equations [40, 48]:

- Data = signs + syntax.
- Information = data + meaning.
- Knowledge = information (syntax and semantics) + ability to use information.

The KDD process is performed within a KDD system that is composed of the following elements: the databases, the either symbolic or numerical data mining modules, and the interfaces for interactions with the system, e.g. editing and visualization. Moreover, the KDD system may take advantage of domain knowledge embedded within an *ontology* relative to the data domain. Closing the loop, the knowledge units extracted by the KDD system must be represented in an adequate representation formalism and then they may be integrated within the ontology to be reused for problem-solving needs in application domains such as agronomy, biology, chemistry, medicine...

This research report is a smooth introduction to KDD that will focus on the so-called symbolic methods in knowledge discovery. There are a number of general books that

can be used with profit for understanding the KDD principles and the usage of the KDD methods, historical research books such as [17, 34], and more recent textbooks such as [24, 25, 14], and [49] that is associated with the Weka system[1]. In the following, we present three symbolic methods for KDD, namely lattice-based classification, frequent itemset search and association rule extraction. Then, we detail some applications of the KDD process, and we propose a discussion and a conclusion for ending the report.

2 Methods for KDD

2.1 An introducting example

Firstly, let us examine what may be expected from the application of data mining methods to data. Let us consider a Boolean matrix $\mathtt{M_{ij}}$, also called a *formal context*, where the rows materialize *customers*, and the columns *products* bought by the customers (see figure2): $\mathtt{M_{ij}} = \mathtt{1}$ whenever the customer `i` buys the product `j`. In real-world formal contexts, such a Boolean matrix may have thousands of columns, and millions of lines... From this formal context, one may extract the following units:

- The set $\mathtt{X} = \{\mathtt{beer}, \mathtt{sausage}, \mathtt{mustard}\}$ has a frequency $\phi(\mathtt{X}) = \mathtt{0.4}$, i.e. there are four individuals on ten buying the three products together. In the same way, the set $\mathtt{X'} = \{\mathtt{beer}, \mathtt{sausage}\}$ has a frequency $\phi(\mathtt{X'}) = \mathtt{0.6}$. The set `X` (respectively `X'`) may be interpreted as the fact that 40% (resp. 60%) of the customers buy the products in `X` (resp. in `X'`) at the same time.
- Moreover, the rule $\mathtt{R} = \{\mathtt{beer\ and\ sausage} \longrightarrow \mathtt{mustard}\}$ may be extracted from the sets `X` and `X'` (i.e. $\mathtt{X'} \longrightarrow \mathtt{X} \setminus \mathtt{X'}$ where $\mathtt{X} \setminus \mathtt{X'}$ denotes the set `X` without `X'`)), with the confidence `0.66`, i.e. if a customer buys `sausage` and `beer`, then the probability that he buys `mustard` is `0.66` (among six customers buying sausage and beer, four customers are also buying mustard).

  From the point of view of the analyst, the sets `X` and `X'`, and the rule `R` as well, may be interpreted and validated as knowledge units extracted from the data.

2.2 Data mining methods

The extraction process is based on *data mining methods* returning knowledge units from the considered data. The data mining methods can be either symbolic or numerical:

- Symbolic methods include among others: classification based on decision trees, lattice-based classification, frequent itemsets search and association rule extraction, classification based on rough sets [39], learning methods, e.g. induction, instance-based learning, explanation-based learning [35, 34], and database methods based on information retrieval and query answering...

[1] `http://www.cs.waikato.ac.nz/ ml/weka/`

Customers/Products	chips	mustard	sausage	soft drink	beer
C_1	1	0	0	0	1
C_2	1	1	1	1	1
C_3	1	0	1	0	0
C_4	0	0	1	0	1
C_5	0	1	1	1	1
C_6	1	1	1	0	1
C_7	1	0	1	1	1
C_8	1	1	1	0	0
C_9	1	0	0	1	0
C_{10}	0	1	1	0	1

Figure 2: An example of a Boolean matrix representing transactions between customers (C) and products (P).

- Numerical methods include among others: statistics and data analysis, hidden Markov models of order 1 and 2 (initially designed for pattern recognition), bayesian networks, neural networks, genetic algorithms...

These methods are dependent on research domains to which the KDD process is linked [32]:

- *Statistics and data analysis*: the goal is similar, but the KDD process requires most of the time a combination of different methods, symbolic as well as numerical methods, and domain knowledge for the interpretation of the extracted units.
- *Database management*: database management system techniques may be used to help the data mining task, e.g. using the query capabilities for preparing data to be mined.
- *Machine learning*: machine learning methods are the core of the KDD process, but scalability, i.e. the amount of data that is considered, and the objectives are different, i.e. reusing the results of the KDD process for problem-solving or decision taking.
- *Knowledge representation and reasoning*: the data mining process may be guided by a model –a domain ontology– for interpretation and problem-solving.

The KDD process may be considered as a kind of "supervised learning process"– since an analyst is in charge of controlling and guiding the KDD process. The analyst may take advantage of his own knowledge and of domain ontologies, for giving an interpretation of the results and for validating the results. In this way, the results of the KDD process may be reused for enlarging existing ontologies, showing that knowledge representation and KDD are two complementary processes: *no data mining without knowledge on the data domain!*

In the following, we are mainly interested in symbolic KDD methods based on the *classification* operation, more precisely on lattice-based classification, frequent itemset search, and association rule extraction. We show how the whole transformation process from rough data into knowledge units is based on the underlying idea of *classification*. Classification is a polymorphic procedure involved in every step of the KDD process: within the mining process, the modeling of the domain for designing a domain ontology, and within domain knowledge representation and reasoning as well.

3 Lattice-based classification

A number of classification problems can be formalized by means of a class of individuals (or objects) and a class of properties (or attributes), and a binary correspondence between the two classes, indicating for each individual-property pair whether the property applies to the individual or not [4, 23, 21]. The properties may be features that are present or absent, or the values of a property that have been dichotomized into Boolean variables. These variables are collected into Boolean tables relating a set of individuals with a set of properties, where $(\mathtt{i}, \mathtt{j}) = \mathtt{1}$ or is *true* whenever the individual $\mathtt{i}$ has the property $\mathtt{j}$ (just as illustrated by the figure 2).

Lattice-based classification relies on the analysis of such Boolean tables and may be considered as a symbolic data mining technique that can be used for extracting from a database a set of concepts organized within a hierarchy (i.e. a partial ordering), frequent itemsets, i.e. sets of properties or features of data occurring together with a certain frequency, and association rules with a given confidence emphasizing correlations between sets of properties.

More precisely, a lattice is an ordered set $(\mathtt{E}, \sqsubseteq)$, where $\sqsubseteq$ denotes a partial ordering such that every pair of elements $(\mathtt{x}, \mathtt{y})$ has an *upper bound* $\mathtt{x} \vee \mathtt{y}$ and a *lower bound* $\mathtt{x} \wedge \mathtt{y}$ [12]. The power-set $2^{\mathtt{E}}$ of a set $\mathtt{E}$ equipped with the inclusion relation is a basic example of a lattice (see figure 3). The set of natural numbers $\mathbb{N}$ equipped with the divisibility relation is also a lattice: $\mathtt{x} \sqsubseteq \mathtt{y}$ if and only if $\mathtt{y}$ is a divisor of $\mathtt{x}$ in $\mathbb{N}$ (see figure 4).

A lattice may be built according to the so-called *Galois* correspondence, classifying within a formal concept a set of individuals, i.e. the *extension* of the concept, sharing a same set of properties, i.e. the *intension* of the concept. Considering the Boolean correspondence between individuals and properties (as shown in figure 2), it is possible to derive for each individual $\mathtt{i}$ the set of all properties that apply to $\mathtt{i}$. Similarly, it is possible to derive for each property $\mathtt{j}$ the set of all individuals to which $\mathtt{j}$ applies. One may further derive *rectangles*, i.e. pairs $\mathtt{O} \times \mathtt{A}$ where $\mathtt{O}$ is a set of individuals and $\mathtt{A}$ a set of properties, such that every property of $\mathtt{A}$ applies to every individual of $\mathtt{O}$. Moreover, *maximal rectangles* $\mathtt{O} \times \mathtt{A}$ are such that the property set $\mathtt{A}$ consists of all common properties of the individuals in $\mathtt{O}$, and that the individual set $\mathtt{O}$ consists of *all* individuals to which the properties of $\mathtt{A}$ jointly apply. Maximal rectangles are called *formal concepts*: they are *concepts* because they actually represent a class of objects, where the individual set $\mathtt{O}$ is the *extension* of the class, and the property set $\mathtt{A}$

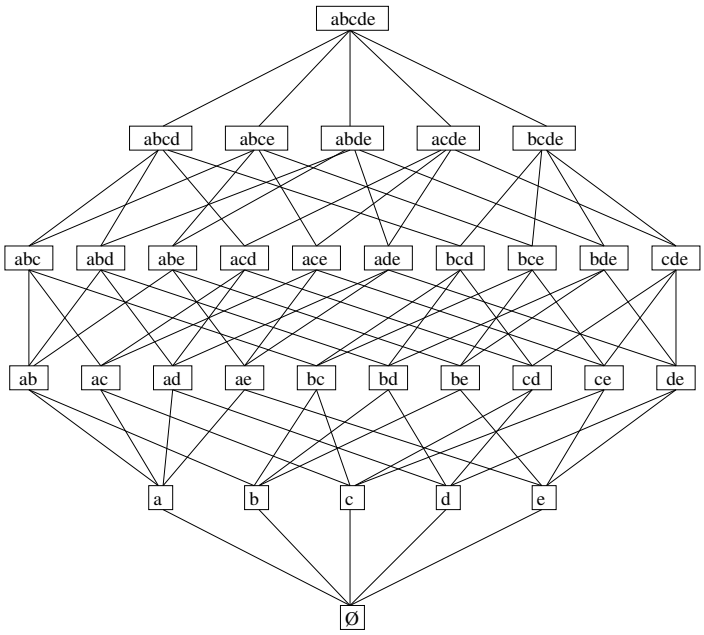

Figure 3: The lattice representing the power set of the set {a,b,c,d,e}.

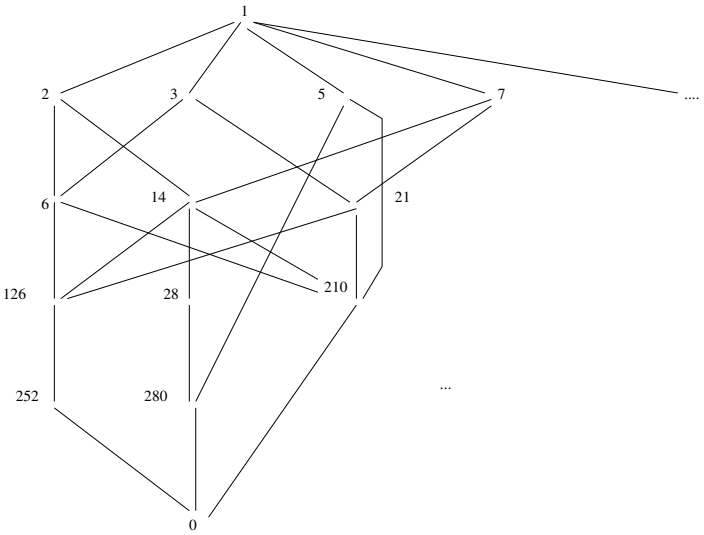

Figure 4: The lattice representing a part of the divisibility relation in $\mathbb{N}$.

is the *intension* of the class ; they are *formal concepts* because they are mathematical entities that do not necessarily refer to any reality.

From a mathematical point of view, let E and F be two finite sets, and R a binary correspondence on $\mathtt{E} \times \mathtt{F}$.

Definition 1 *The mapping* $\mathtt{f} : \mathtt{E} \longrightarrow \mathtt{F}$ *is such that, if* x *is an element of* E, $\mathtt{f}(\{\mathtt{x}\})$ *consists of all elements of* F *related to* x *by* R. *If* X *is an arbitrary part of* E, $\mathtt{f}(\mathtt{X}) = \{\mathtt{y} \in \mathtt{F} / \forall \mathtt{x} \in \mathtt{X} : \mathtt{xRy}\}$.

Dually, the mapping $\mathtt{g} : \mathtt{F} \longrightarrow \mathtt{E}$ *is such that, if* y *is an element of* F, $\mathtt{g}(\{\mathtt{y}\})$

Objects / Items	a	b	c	d	e
O_1	0	1	1	0	1
O_2	1	0	1	1	0
O_3	1	1	1	1	0
O_4	1	0	0	1	0
O_5	1	1	1	1	0
O_6	1	0	1	1	0

Figure 5: An example of formal context.

consists of all elements of E *that are related to* y *by* R. *If* Y *is an arbitrary part of* F, $g(Y) = \{x \in E / \forall y \in Y : xRy\}$.

The couple $\{f, g\}$ *is said to be a* Galois connection *or a* Galois correspondence *between the sets* E *and* F.

In terms of objects and attributes, $f(X)$ is the set of all attributes shared by all objects in X, and $g(Y)$ is the set of all objects that have all attributes of Y. Moreover, $X \subseteq X' \Rightarrow f(X') \subseteq f(X)$, and $Y \subseteq Y' \Rightarrow g(Y') \subseteq g(Y)$: the mappings f and g are decreasing. For example, considering the Boolean table of the figure 5, we have $f(\{O_1\}) = \{b, c, e\}$ and $g(\{b, c, e\}) = \{O_1\}$, $f(\{O_1, O_2\}) = \{c\}$ and $g(\{c\}) = \{O_1, O_2, O_3, O_5, O_6\}$, $g(\{a, c\}) = \{O_2, O_3, O_5, O_6\}$ and $f(\{O_2, O_3, O_5, O_6\}) = \{a, c, d\}$.

The mapping $h = g \circ f = g[f]$ maps every part of E onto a part of E, and the mapping $h' = f \circ g = f[g]$ maps every part of F onto a part of F. It can be shown that the mappings h and h' are *closure operators*:

Definition 2 *A* closure operator h *is: (i) monotonously increasing, i.e. if* X *and* X' *are subsets of* E*:* $X \subseteq X' \Rightarrow h(X) \subseteq h(X')$*, (ii) extensive, i.e.* $X \subseteq h(X)$*, and (iii) idempotent, i.E.* $h(X) = h[h(X)]$.

A subset X *of* E *is said to be* closed *if and only if* $X = h(X)$.

The closure operators $h = g \circ f = g[f]$ for E and $h' = f \circ g = f[g]$ for F are said to be *Galois closures*. Let L_E and L_F be the sets of all closed parts of E and F respectively, partially ordered by set inclusion. Then, $(L_E, \subseteq)$ and $(L_F, \subseteq)$ have lattice structures: the meet of two parts is their intersection, whereas the join of two parts is the closure of their union[2]. The Galois connection $\{f, g\}$ restricted to the closed parts of E and F materializes a one-to-one correspondence between the lattices $(L_E, \subseteq)$ and $(L_F, \subseteq)$.

We may now consider the set L of all couples of corresponding parts of L_E and L_F, i.e. each element of L is the Cartesian product of closed parts of E and F, denoted by $(X, f(X))$, or $(g(Y), Y)$, with X, $f(X)$, Y, and $g(Y)$ being closed. The partial order relation $\sqsubseteq$ may be defined on L such that $(X, Y) \sqsubseteq (X', Y')$ if and only if $X' \subseteq X$ (or dually $Y \subseteq Y'$. The structure $(L, \sqsubseteq)$ is the *Galois lattice* or the *concept lattice* of the

[2]The union of two closed sets is not necessarily a closed set as it is the case for the intersection of two closed sets.

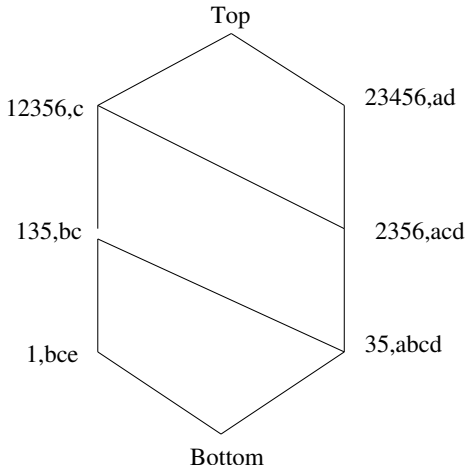

Figure 6: The Galois lattice associated to the formal context.

relation R on $\mathtt{E} \times \mathtt{F}$, and it can be demonstrated that the elements of L are the formal concepts derived from the relation R. For example, the Galois lattice associated to the formal context introduced on figure 5 is shown on figure 6.

More precisely, the partial order between two concepts $(\mathtt{X}, \mathtt{Y}) \sqsubseteq (\mathtt{X}', \mathtt{Y}')$ verifies that the extension $\mathtt{X}'$ of $(\mathtt{X}', \mathtt{Y}')$, i.e. the *subsumer* concept, includes the extension X of $(\mathtt{X}, \mathtt{Y})$, i.e. the *subsumee* concept, and, dually, that the intension $\mathtt{Y}'$ of $(\mathtt{X}', \mathtt{Y}')$ is included in the intension Y of $(\mathtt{X}, \mathtt{Y})$. Thus, there exists an order-reversing one-to-one correspondence between the extensions and the intensions of formal concepts, *covariant* for the extensions and *contravariant* for the intensions. Moreover, there exists a number of algorithms for building Galois lattices –see [22, 15, 21, 29, 30]– with different and specific characteristics.

The notion of Galois lattice has given rise to the so-called *lattice-based classification*, and to the active research domain of *formal concept analysis*[3] [21]. Formal concept analysis is used for a number of different tasks, among which the design of object hierarchies, especially in object-oriented programming, for designing class hierarchies. Furthermore, lattice-based classification may be used for a number of purposes in KDD [43, 48, 45]:

- Since the concepts are the basic units of human thought (and hence the basic structures of logic), the logical structure of information is based on concepts and concept systems. Therefore, Galois or concept lattices as mathematical abstraction of concept systems can support humans to discover information and then to create knowledge.

- It is important to have a mathematization of concepts that reflects the rich logical functionalities in which concepts are embedded in the real-world. Concept lattices and lattice-based classification are examples of such mathematical tools.

[3] fca-list@aifb.uni-karlsruhe.de
http://www.aifb.uni-karlsruhe.de/mailman/listinfo/fca-list

Indeed, the mathematical structure of a concept lattice is effectively accessible to human reasoning by labeled line diagrams (lattice drawings).

- Lattice-based classification and formal concept analysis is a suitable paradigm for KDD, as discussed in [45]. The mathematical and algorithmic backgrounds exist and may be used for real-sized applications [28, 30]. Moreover, some improvements may be carried on, especially on facility, i.e. the ease of use of the data mining methods, on the cost-effectiveness of the methods allowing effective and efficient implementations, e.g. distributive and parallel architectures, and finally on adaptability, i.e. the ability to fit evolving situations with respect to the constraints that may be associated to the KDD process. Moreover, one other major research point is the extension of lattice-based classification to complex objects, where properties may be multi-valued properties, or even relations.

4 Frequent itemset search and association rule extraction

In parallel with lattice-based classification, one may extract frequent itemsets and association rules from data (as shown in the introductory example in § 2). The extraction of frequent itemsets consists in extracting from formal Boolean contexts sets of properties occurring with a support, i.e. the number of individuals sharing the properties, that must be greater than a given threshold. From the frequent itemsets, it is then possible to generate association rules of the form $\mathtt{A} \longrightarrow \mathtt{B}$ relating a subset of properties A with a subset of properties B, that can be interpreted as follows: the individuals including A include also B with a certain support and a certain confidence. The numbers of itemsets and rules that can be extracted from a formal Boolean context may be very large, and thus there is a need for pruning the sets of itemsets and the sets of extracted rules for ensuring a subsequent interpretation of the extracted units. This is especially true when the interpretation has to done –and this is usually the case– by an analyst who is in charge of interpreting the results of the KDD process.

In the following, we introduce the principles of frequent itemset search and of the extraction of association rules. Then practical examples of both processes are proposed.

4.1 Frequent itemset search

Definition 3 *Given a set of objects* O *and a set of properties* P*, an* item *corresponds to a property of an object, and an* itemset*, or a* pattern*, to a set of items: an object is said to own an item. The number of items in an itemset determines the* length *of the itemset. The* image *of an itemset corresponds to the set of objects owning the item.*

The support *of an itemset corresponds to the proportion of objects owning the itemset, with respect to the whole population of objects. An itemset is said to be* frequent *if its support is greater than a given* frequency threshold $\sigma_{\mathtt{s}}$*: a proportion at*

least equal to $\sigma_{\mathtt{s}}$ *of objects own* all *items included in the itemset.*

For example, let us consider the formal context introduced on figure 5, with $\sigma_{\mathtt{s}} = 3/6$, we have: {a} is a frequent itemset of length 1 and of support 5/6; {ac} is of length 2, of support 4/6, and frequent; {abc} is of length 3, of support 2/6, and not frequent; {abcde} is of length 5, of support 0/6, and not frequent. It can be noticed that the support is a monotonously decreasing function, with respect to the length of an itemset.

When the number of properties in P is equal to n, the number of potential itemsets is equal to 2^n (actually, the number of all possible subsets of the set P): thus, a direct search for the frequent itemsets by directly testing the itemsets that are frequent is not conceivable. Heuristics have to be used for pruning the set of all itemsets to be tested. This is the purpose of the so-called *level-wise search* of frequent itemsets, and the associated well-known Apriori algorithm [1, 3, 33, 2]. The Apriori algorithm relies on two fundamentals and dual principles: *(i) every sub-itemset of a frequent itemset is a frequent itemset, (ii) every super-itemset of a non frequent itemset is non frequent.* The Apriori algorithm can be summarized as follows:

1. The search of frequent begins with the search of frequent itemsets of length 1.
2. The frequent itemsets are recorded and combined together to form the *candidate* itemsets of greater length. The non-frequent itemsets are discarded, and by consequence, all their super-itemsets. The candidate itemsets are then tested, and the process continues in the same way, until no more candidates can be formed.

For example, considering the formal context on figure 5, with $\sigma_{\mathtt{s}} = 2/6$, the frequent itemsets of length 1 are {a} (3/6), {b} (5/6), {c} (5/6), {d} (5/6). The itemset {e} (1/6) is not frequent and pruned. Then the candidates of length 2 are formed, combining the frequent itemsets of length 1, e.g. {ab}, {ac}, {ad}... and then tested. The frequent itemsets of length 2 are {ab} (2/6), {ac} (4/6), {ad} (5/6), {bc} (3/6), {bd} (2/6), {cd} (4/6). The candidates of length 3 are formed and tested: the frequent itemsets of length 3 are {abc} (2/6), {abd} (2/6), {acd} (4/6), {bcd} (2/6). Finally, the candidate of length 4 is formed, i.e. {abcd}, tested and recorded as a frequent itemset ({abcd} (2/6)). No other candidates can be formed, and the algorithm terminates.

When the data to be mined are huge, i.e. millions of rows and thousands of columns, there is a need for minimizing the access to the data for calculating the support. A number of studies have been carried out in this direction, giving rise to very efficient algorithms (see for example [38, 37, 42, 50]).

Lattices and itemsets are related: actually, the search for frequent itemsets corresponds to a breadth-first search in the concept lattice associated to the formal context under study. However, an itemset corresponds to a subset of properties, without being necessarily a closed set. In this way, the property of closure for an itemset is one of the characteristics on which rely fast algorithms searching for itemsets (it can be noticed that the name of one of these algorithms is Close [38, 37]).

4.2 Association rule extraction

Definition 4 *An* association rule *has the form* $\mathtt{A} \longrightarrow \mathtt{B}$*, where* $\mathtt{A}$ *and* $\mathtt{B}$ *are two itemsets. The* support *of the rule* $\mathtt{A} \longrightarrow \mathtt{B}$ *is defined as the support of the itemset* $\mathtt{A} \sqcap \mathtt{B}$ *(where* $\sqcap$ *denotes the union of itemsets). The* confidence *of a rule* $\mathtt{A} \longrightarrow \mathtt{B}$ *is defined as the quotient* $\mathtt{support}(\mathtt{A} \sqcap \mathtt{B})/\mathtt{support}(\mathtt{A})$*. The confidence can be seen as a conditional probability* $\mathtt{P}(\mathtt{B}/\mathtt{A}) = \mathtt{support}(\mathtt{A} \sqcap \mathtt{B})/\mathtt{support}(\mathtt{A})$ *i.e. probability of* $\mathtt{B}$ *knowing* $\mathtt{A}$*.*

A rule is said to be valid *if its confidence is greater than a* confidence threshold $\sigma_{\mathtt{c}}$*, and its support is greater than the* frequency threshold *for itemsets* $\sigma_{\mathtt{s}}$ *(a valid rule can only be extracted from a frequent itemset). A rule is said to be* exact *if its confidence is of* $\mathtt{1}$*, i.e.* $\mathtt{support}(\mathtt{A} \sqcap \mathtt{B}) = \mathtt{support}(\mathtt{A})$*, otherwise the rule is* partial*.*

For example, with $\sigma_{\mathtt{s}} = 3/6$ and $\sigma_{\mathtt{c}} = 3/5$, $\{\mathtt{ac}\}$ is frequent, and the rule $\mathtt{a} \longrightarrow \mathtt{c}$ is valid (with support 4/6 and confidence 4/5); the rule $\mathtt{c} \longrightarrow \mathtt{a}$ is valid (with support 4/6 and confidence 4/5). With $\sigma_{\mathtt{s}} = 2/6$ and $\sigma_{\mathtt{c}} = 3/5$, $\{\mathtt{abd}\}$ is frequent, the rule $\mathtt{b} \longrightarrow \mathtt{ad}$ is valid (with support 2/6 and confidence 2/3); the rule $\mathtt{ad} \longrightarrow \mathtt{b}$ is not valid (with support 2/6 and confidence 2/5).

The generation of association valid rules from a frequent itemset (of length necessarily greater or equal to 2) proceeds in a similar way as the search for frequent itemsets. Given a frequent itemset $\mathtt{P}$, the extraction starts by generating the valid rules with a conclusion of length 1, say rules of the form $\mathtt{P} \setminus \{\mathtt{i}\} \longrightarrow \{\mathtt{i}\}$, where $\{\mathtt{i}\}$ is an item of length $\mathtt{1}$, and $\mathtt{P} \setminus \{\mathtt{i}\}$ denotes the itemset $\mathtt{P}$ without the item $\{\mathtt{i}\}$. Then, the conclusions of the valid rules $\mathtt{P} \setminus \{\mathtt{i}\} \longrightarrow \{\mathtt{i}\}$ are combined for generating the candidate conclusions of length 2, e.g. $\mathtt{P} \setminus \{\mathtt{ij}\} \longrightarrow \{\mathtt{ij}\}$, and the process continues until no more valid rules can be generated from the frequent itemset.

For example, with our current formal context, given $\sigma_{\mathtt{s}} = 2/6$ and $\sigma_{\mathtt{c}} = 2/5$, when $\mathtt{P} = \{\mathtt{ab}\}$, the generated valid rules are $\{\mathtt{a}\} \longrightarrow \{\mathtt{b}\}$ (2/6,2/5) and $\{\mathtt{b}\} \longrightarrow \{\mathtt{a}\}$ (2/6,2/3). Given the frequent itemset $\mathtt{P} = \{\mathtt{abc}\}$ (2/6), the generated rules are $\{\mathtt{ab}\} \longrightarrow \{\mathtt{c}\}$ (2/6,1), $\{\mathtt{ac}\} \longrightarrow \{\mathtt{b}\}$ (2/6,1/2), $\{\mathtt{bc}\} \longrightarrow \{\mathtt{a}\}$ (2/6,2/3); as $\{\mathtt{a,b,c}\}$ are three valid conclusions, they can be combined for producing the new conclusions $\{\mathtt{ab,ac,bc}\}$, and generate the rules $\{\mathtt{c}\} \longrightarrow \{\mathtt{ab}\}$ (2/6,2/5), $\{\mathtt{b}\} \longrightarrow \{\mathtt{ac}\}$ (2/6,2/3), $\{\mathtt{a}\} \longrightarrow \{\mathtt{bc}\}$ (2/6,2/5), which are all valid rules.

There exists a number of studies on the possible measures that can be attached to an association rule [31, 44, 10]. Considering the confidence of the rule $\mathtt{A} \longrightarrow \mathtt{B}$ as the conditional probability $\mathtt{Prob}(\mathtt{B}/\mathtt{A})$ (probability of $\mathtt{B}$ knowing $\mathtt{A}$), other measures may be built on the basis of probability calculus:

- The *interest* or *lift* of the rule $\mathtt{A} \longrightarrow \mathtt{B}$ measure is defined as $\mathtt{Prob}(\mathtt{A} \sqcap \mathtt{B})/\mathtt{Prob}(\mathtt{A}) \times \mathtt{Prob}(\mathtt{B})$, i.e. the interest measures the degree of compatibility of $\mathtt{A}$ and $\mathtt{B}$, i.e. the simultaneous occurrences of both events $\mathtt{A}$ and $\mathtt{B}$.

- The *conviction* of the rule $\mathtt{A} \longrightarrow \mathtt{B}$ is defined as $\mathtt{Prob}(\mathtt{A}) \times \mathtt{P}(\neg\mathtt{B})/\mathtt{P}(\mathtt{A} \sqcap \neg\mathtt{B})$, i.e. the conviction measures the deviation of the rule $\mathtt{A} \longrightarrow \mathtt{B}$ from the rule $\mathtt{A} \longrightarrow \neg\mathtt{B}$, or, in other word, how is high the degree of implication of the rule $\mathtt{A} \longrightarrow \neg\mathtt{B}$.

- The *dependency* of the rule $\mathtt{A} \longrightarrow \mathtt{B}$ is defined as —Prob(B/A) - Prob(B)— $= |(\mathtt{Prob}(\mathtt{A} \sqcap \mathtt{B}) - \mathtt{Prob}(\mathtt{A}) \times \mathtt{Prob}(\mathtt{B}))/\mathtt{Prob}(\mathtt{A})|$, i.e. the dependency measures the degree of independence between the events A and B, i.e. the fact that the occurrence of the event A is or is not not dependent on the occurrence of the event B.

In the same way as lattice-based classification, frequent itemset search and association rule extraction may be used with benefit for KDD tasks. In the following, we present two real-world applications where these two data mining methods have been successfully applied on real world data.

5 Applications

In the following, we detail three applications of the KDD process relying on the data mining techniques presented here-above: an experiment in mining reaction databases for organic chemistry planning, an application in mining gene expression databases in biology, and an introduction to Web mining that concludes the section.

5.1 Mining chemical reaction database

In this paragraph, we present an experiment on the application of knowledge discovery algorithms for mining chemical reaction databases [6, 5]. Chemical reactions are the main elements on which relies synthesis in organic chemistry, and this is why chemical reaction databases are of first importance. Synthesis planning is mainly based on *retrosynthesis*, i.e. a goal-directed problem-solving approach, where the target molecule is iteratively transformed by applying reactions for obtaining simpler fragments, until finding accessible starting materials (see figure 7). For a given target molecule, a huge number of starting materials and reactions may exist, e.g. thousands of commercially available chemical compounds. Thus, exploring all the possible pathways issued from a target molecule leads to a combinatorial explosion, and needs a strategy for choosing reaction sequences to be used within the planning process.

From a problem-solving process perspective, synthesis in organic chemistry must be considered at two main levels of abstraction: a *strategic* level, where general synthesis methods are involved, and a *tactic* level, where actual chemical reactions are applied. The present experiment is aimed at discovering generic reactions, also called *synthesis methods*, from chemical reaction databases in order to design generic and reusable synthesis plans. This can be understood in the following way: mining reaction databases at the tactic level for finding synthesis methods at the strategic level. This knowledge discovery process relies on the one hand on mining algorithms, i.e. frequent levelwise itemset search and association rule extraction, and, on the other hand, on domain knowledge, that is involved at every step of the knowledge discovery process.

At present, reaction database management systems are the most useful tools for helping the chemist in synthesis planning. One aspect of the present experiment is

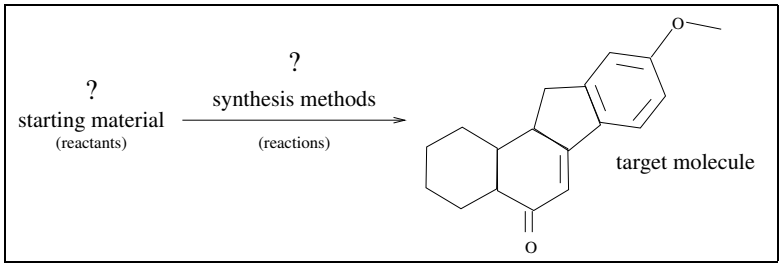

Figure 7: The general schema of a synthesis problem.

to study how data mining techniques may contribute to knowledge extraction from reaction databases, and beyond that, to the structuring of these databases and the improvement of the database querying. Two reaction databases have been mined using frequent itemset search and association rule extraction. This experiment is original and novel within the domain of organic synthesis planning. Regarding the knowledge discovery research, this experiment stresses the fact that knowledge extraction within a complex application domain has to be guided by knowledge domain if substantial results have to be obtained.

5.1.1 The chemical context

Actually, the main questions for the synthesis chemist are related to chemical families to which belongs a target molecule, i.e. the molecule that has to be built, and to the reactions or sequence of reactions building structural patterns, to be used for building these families. Two main categories of reactions may be distinguished: reactions building the *skeleton* of a molecule –the arrangement of carbon atoms on which relies a molecule–, and reactions changing the *functionality* of a molecule, i.e. changing a function into another function (see figure 8). Hereafter, we are mainly interested in reactions changing the functionality, and especially in the following question: what are the reactions allowing the transformation of a function $\mathtt{F_i}$ into a function $\mathtt{F_j}$?

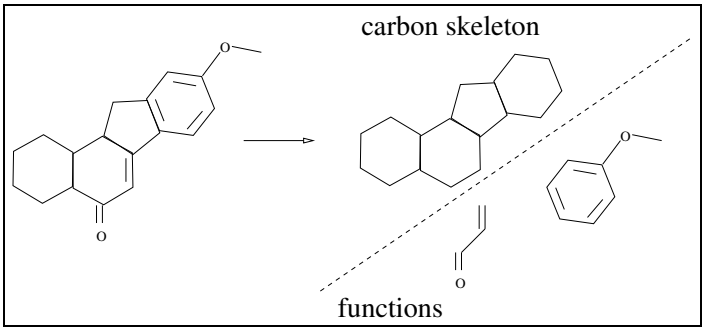

Figure 8: Skeleton and functional groups of a target molecule.

The experiment reported hereafter has been carried out on two reaction databases, namely the "Organic Syntheses" database ORGSYN-2000 including 5486 records, and the "Journal of Synthetic Methods" database JSM-2002 including 75291 records. The information items in databases such as ORGSYN-2000 and JSM-2002 may be seen as

a collection of records, where every record contains one chemical equation involving structural information, that can be read, according to the reaction model, as the transformation of an *initial state* –or the set of *reactants*– into a *final state* –or the set of *products*– associated with an atom-to-atom mapping between the initial and final states (see fig. 9).

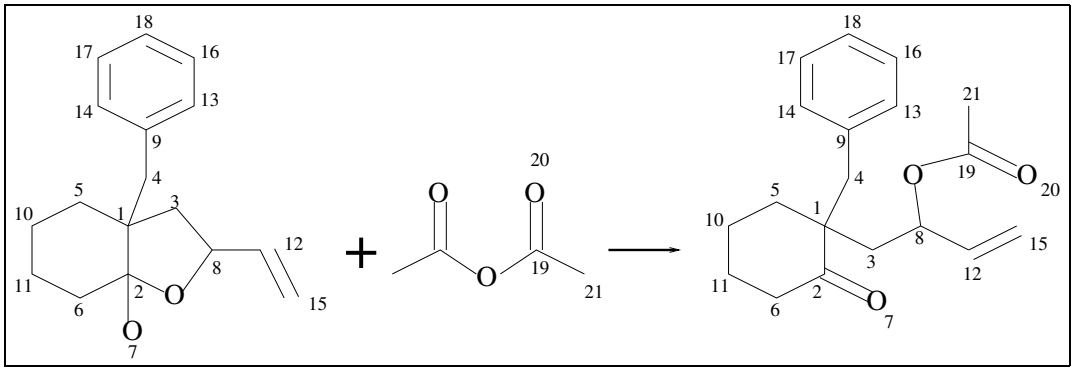

Figure 9: The structural information on a reaction with the associated atom-to-atom mapping (reaction #13426 in the JSM-2002 database).

The purpose of the preprocessing step of data mining is to improve the quality of the selected data by cleaning and normalizing the data. In this framework, data preprocessing has mainly consisted in exporting and analyzing the structural information recorded in the databases for extracting and for representing the functional transformations in a target format that has been processed afterwards. The considered transformations are functional modifications, functional addition and deletion, i.e. adding or deleting a function. The reactions have been considered at an abstract level, the so-called *block level* as shown in figure 10. The transformation of a reaction at the block level is carried out thanks to the RESYN-ASSISTANT knowledge system [46, 36], whose objective is to help synthesis problem-solving in organic chemistry. This points out the role of knowledge and knowledge systems within the KDD process.

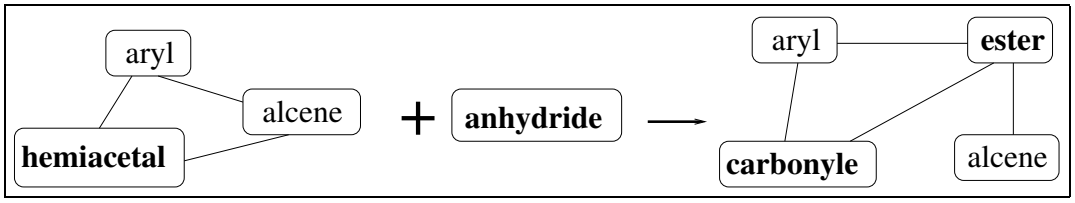

Figure 10: The representation of the reaction #13426 in the JSM-2002 database at the block level.

5.1.2 Mining of a reaction database

The RESYN-ASSISTANT system [46] has been used for recognizing the building blocks of reactions. Based on the atom-to-atom mapping, the system establishes the correspondence between the recognized blocks of the same nature, and determines their

Entries/Blocks	**Destroyed**		**Formed**		**Unchanged**	
	anhydride	hemiacetal	carbonyle	ester	alcene	aryle
without correspondence entry **R**	x	x	x	x	x	x
with correspondence entry $\mathbf{R}_1$	x	x		x	x	x
entry $\mathbf{R}_2$		x	x		x	x

Figure 11: The original data are prepared for the mining task: the Boolean transformation of the data can be done without taking into account the atom mapping, i.e. one single line in the Boolean table, or by taking into account the atom mapping, i.e. two lines in the table.

role in the reaction. A function may be present in a reactant, in a product, or in both. In the last case, the function is unchanged. In the two other cases, the function in the reactant is destroyed, or the function in the product is formed. During a reaction, either one or more reactant functions may contribute to form the functions in the products. At the end of the preprocessing step, the information obtained by the recognition process is incorporated into the representation of the reaction.

For allowing the application of the algorithms for frequent itemset search and association rule extraction, namely the Close algorithm [38, 37], the data on reactions have been transformed into a Boolean table (loosing the actual representation of a molecule as a composition of functional blocks). Then, a reaction can be considered from two main points of view (see figure 11):

- a global point of view on the functionality interchanges leads to consider a single entry $\mathtt{R}$ corresponding to a single analyzed reaction, to which is associated a list of properties, i.e. formed and/or destroyed and/or unchanged functions,
- a specific point of view on the functionality transformations that is based on the consideration of two (or more) different entries $\mathtt{R_k}$ corresponding to the different functions being formed.

Both correspondences have been used during the experiment. The Close algorithm has been applied to Boolean tables for generating first itemsets, i.e. sets of functions (with an associated support), and then association rules. The study of the extracted frequent itemsets may be done with different points of view. Studying frequent itemsets of length 2 or 3 enables the analyst to determine basic relations between functions. For

example searching for a formed functions $\mathtt{F_f}$ ($\mathtt{\_f}$ for formed) deriving from a destroyed function $\mathtt{F_d}$ ($\mathtt{\_d}$ for destroyed) leads to the study of the itemsets $\mathtt{F_d} \sqcap \mathtt{F_f}$, where the symbol $\sqcap$ stands for the conjunction of items or functions. In some cases, a reaction may depend on functions present in both reactants and products that remain unchanged ($\mathtt{\_u}$ for unchanged) during the reaction application, leading to the study of frequent itemsets such as $\mathtt{F_f} \sqcap \mathtt{F_u} \sqcap \mathtt{F_d}$. This kind of itemsets can be searched and analyzed for extracting a "protection function" supposed to be stable under given experimental conditions.

The extraction of association rules gives a complementary perspective on the knowledge extraction process. For example, searching for the more frequent ways to form a function $\mathtt{F_f}$ from a function $\mathtt{F_d}$ leads to the study of rules such as $\mathtt{F_f} \longrightarrow \mathtt{F_d}$: indeed, this rule has to be read in a retrosynthesis way, i.e. if the function $\mathtt{F_f}$ is formed then this means that the function $\mathtt{F_d}$ is destroyed. Again, this rule can be generalized in the following way: determining how a function $\mathtt{F_f}$ is formed from two destroyed functions $\mathtt{F_{d1}}$ and $\mathtt{F_{d2}}$, knowing say that the function $\mathtt{F_{d1}}$ is actually destroyed, leads to the study of the association rules such as $\mathtt{F_f} \sqcap \mathtt{F_{d1}} \longrightarrow \mathtt{F_{d2}}$.

5.1.3 Looking at the extracted itemsets and rules results

A whole set of results of the application of the data mining process on the ORGSYN-2000 and JSM-2002 databases is given in [6]. These results show that both reaction databases share many common points though they differ in terms of size and data coverage, i.e. among 500 functions included in the concept hierarchy of functional graphs within the knowledge base of the RESYN-ASSISTANT system, only 170 are retrieved from ORGSYN-2000 while 300 functions are retrieved from JSM-2002. The same five functions are ranked at the first places in both databases with the highest occurrence frequency. However, some significant differences can be observed: a given function may be much more frequent in the ORGSYN-2000 database than in JSM-2002 database, and reciprocally. These differences can be roughly explained by different data selection criteria and editor motivations for both databases.

A qualitative and statistical study of the results has shown the following behaviors. Some functions have a high stability, i.e. they mostly remain unchanged, and, in the contrary, some others functions are very reactive, i.e. they are mostly destroyed. All the reactive functions are more present in reactants than in products, and some functions are more often formed. Some functions, that are among the most widely used functions in organic synthesis, are more often present and destroyed in reactants, e.g. `alcohol` and `carboxylic acid`. For example, among the standard reactions involving functions, it is well-known –for chemists– that the `ester` function derives from a combination of two functions, one of them being mostly an `alcohol`. The search for a second function relies on the study of rules such as $\mathtt{ester_f} \sqcap \mathtt{alcohol_d} \longrightarrow \mathtt{F_d}$. The main functions that are retrieved are `anhydride`, `carboxylic acid`, `ester`, and `acyl chloride`. If the chemist is interested in the unchanged functions, then the analysis of the rule $\mathtt{ester_f} \sqcap \mathtt{alcohol_d} \sqcap \mathtt{anhydride_d} \longrightarrow \mathtt{F_u}$ gives functions such as `acetal`, `phenyl`, `alkene`, and `carboxylic acid`.

These first results provide a good overview on the function stability and reactivity. They also give partial answers to the question of knowing what are the reactions allowing the transformation of a function $\mathtt{F_i}$ into a function $\mathtt{F_j}$.

5.1.4 Discussion

A number of topics are discussed hereafter regarding this experiment in mining chemical reaction databases. First, it can be noticed that only a few research works hold on the application of data mining methods on reaction databases ; the study on the lattice-based classification of dynamic knowledge units proposed in [20] has been a valuable source of inspiration for the present experiment. The abstraction of reactions within blocks and the separation in three kinds of blocks, namely formed, destroyed, and unchanged blocks, is one of the most original idea in that research work, that is responsible of the good results that have been obtained. This idea of the separation into three families may be reused in other contexts involving dynamic data. However, the transformation into a Boolean table has led to a loss of information, e.g. the connection information on reactions and blocks.

Frequent items or association rules are generic elements that can be used either to index (and thus organize) reactions or to retrieve reactions. Termed in another way, this means that frequent itemsets or extracted association rules may be in certain cases considered as a kind of meta-data giving meta-information on the bases that are under study.

Knowledge is used at every step of the knowledge extraction process, e.g. the coupling of the knowledge extraction process with the RESYN-ASSISTANT system, and domain ontologies such as the function ontologies, the role of the analyst,... Indeed, and this is one of the major lesson of this experiment: the knowledge discovery process in a specific domain such as organic synthesis has to be *knowledge-intensive*, and has to be guided by domain knowledge, and an analyst as well, for obtaining substantial results. The role of the analyst includes fixing the thresholds, and interpreting of the results. The thresholds must be chosen in function of the objectives of the analyst, and in function of the content of the databases (it can be noticed that a threshold of 1% for an item support means that for a thousand of reactions, ten may form a reaction family, and this is not a bad hypothesis).

Moreover, the use of data mining methods such as frequent itemsets search or association rule extraction has proven to be useful, and has provided encouraging results. It could be interesting to test other (symbolic) data mining methods, and mainly relational mining for being able to take into account the structure of molecule for the data mining task [13, 19, 16].

5.2 An experiment in biology

In this paragraph, we present an experiment on the mining of gene expression databases for extracting association rules, based on the article [11] (see also [47] for a recent overview on data mining in bioinformatics). Global gene expression profiling, can be

a valuable tool in the understanding of genes, biological networks, and cellular states. One goal in analyzing expression data is to try to determine how the expression of any particular gene might affect the expression of other genes ; the genes involved in this case could belong to the same biological network. Another goal of analyzing expression data is to try to determine what genes are expressed as a result of certain cellular conditions, e.g. what genes are expressed in diseased cells that are not expressed in healthy cells.

As larger and larger gene expression data sets become available, data mining techniques can be applied to identify patterns of interest in the data. In [11] is detailed an experiment where an Apriori algorithm has been applied for mining association rules from gene expression data, using a set of data of 300 expression profiles for yeast. An example of extracted association rule is the following: $\{\texttt{cancer}\} \longrightarrow \{\texttt{gene A} \uparrow, \texttt{gene B} \downarrow, \texttt{gene C} \uparrow\}$, meaning that, for the data set that has been mined, in most profile experiments where the cells used are cancerous, `gene A` has been measured as being up (highly expressed), `gene B` is down (low expression), and `gene C` is up. In the context of formal databases, a gene expression profile can be thought of a single transaction (corresponding to a row in a Boolean table), and each protein can be thought as an item. A gene expression profile transaction may include the set of genes that are up and the set of genes that are down in the profile. Items in the transaction can also include relevant facts describing the cellular environment. Moreover, in an expression profile each protein is assigned a real value that specifies the relative abundance of that protein in the profiled sample. These protein values have been made discrete for allowing the processing using standard techniques based on Boolean tables.

The extracted association rules that have been considered in the experiment are of the form $\{\texttt{LHS}\} \longrightarrow \{\texttt{RHS}\}$, where $\{\texttt{LHS}\}$, i.e. *left hand side*, is composed of only one item, and $\{\texttt{RHS}\}$, i.e. *right hand side*, may have an arbitrary number of items. It can be noticed that such association rules, where $\{\texttt{LHS}\}$ is composed of only one item are very interesting and are the basis of efficient algorithms for itemset levelwise search [42], as explained hereafter.

Furthermore, such rules may be used to check the validity of other rules as shown below. Let us consider the rule $\texttt{P}_1 \longrightarrow \texttt{P}_2 \setminus \texttt{P}_1$, where $\texttt{P}_1 \subseteq \texttt{P}_2$, then: $\texttt{support}(\texttt{P}_1 \longrightarrow \texttt{P}_2 \setminus \texttt{P}_1) = \texttt{support}(\texttt{P}_1 \cup (\texttt{P}_2 \setminus \texttt{P}_1)) = \texttt{support}(\texttt{P}_2)$. If $\texttt{P}_1 \longrightarrow \texttt{P}_2 \setminus \texttt{P}_1$ is a valid rule, then $\texttt{P}_2$ has to be a frequent itemset, and $\texttt{P}_1$, as a subset of $\texttt{P}_2$, has to be frequent too. Then, any rule of the form $\texttt{P}'_1 \longrightarrow \texttt{P}_2 \setminus \texttt{P}'_1$, where $\texttt{P}_1 \subseteq \texttt{P}'_1 \subseteq \texttt{P}_2$ is valid too. For example, knowing that $\{\texttt{ab}\} \longrightarrow \{\texttt{cd}\}$ is valid, it can be deduced that $\{\texttt{abc}\} \longrightarrow \{\texttt{d}\}$ and $\{\texttt{abd}\} \longrightarrow \{\texttt{c}\}$ are valid too. This shows that the less is the length of the condition of an association rule of the form $\texttt{P}_1 \longrightarrow \texttt{P}_2 \setminus \texttt{P}_1$, the more we can deduce the validity of rules of the form $\texttt{P}'_1 \longrightarrow \texttt{P}_2 \setminus \texttt{P}'_1$, with $\texttt{P}_1 \subseteq \texttt{P}'_1 \subseteq \texttt{P}_2$. In [42], minimal left hand sides of the rules are generators, and maximal right hand sides of the rules correspond to the closed itemsets related with closed sets of properties constituting the intension of the concepts in the associated lattice.

Actually, in [11], closed itemsets have been mainly considered, and the set of extracted association rules has been manually pruned for a better understandability of the results. In particular, this shows the importance of presenting small sets of as-

sociation rules or frequent itemsets for a valuable human analysis of the results (as discussed in [10] for example). Two examples of extracted association rules are the following, where the minimum support has been fixed to 10%, and the minimum confidence to 80%, and where a rule may be interpreted as follows: when the gene in {LHS} is up, so are the genes in {RHS}. The expressions of rules have been simplified for a better readability for non-biologists.

$\{\texttt{YHM1}\} \longrightarrow \{\texttt{SEQ1}, \texttt{ARO3}\}$

$\{\texttt{ARO3}\} \longrightarrow \{\texttt{SEQ1}, \texttt{YHM1}\}$

where $\{\texttt{SEQ1}\} = \{\texttt{ARG1}, \texttt{ARG4}, \texttt{CTF13}, \texttt{HIS5}, \texttt{LYS1}, \texttt{RIB5}, \texttt{SNO1}, \texttt{SNZ1}, \texttt{YHRO29C}, \texttt{YOL118C}\}$,

An analysis that may be of interest is the following. The genes {YHM1} in the first rule and {ARO3} in the second rule are found on opposite sides of the rules. The gene {YHM1} has been identified as a suppressor of a gene having the property of being a binding factor. On the other hand, the gene {ARO3} is activated by a binding factor itself. Whether the nature of the association suggested here between {ARO3} and {YHM1} has something to do with the fact that both of these genes have an association with a binding factor is an open –and very interesting– question...

The association rules that have been mined represent only a fraction of all the possible gene-to-gene interactions that remain to be discovered in yeast. More rules can be found using different search criteria, i.e. changing the support, the confidence, the data, and the form of the extracted rules. The extracted association rules can lead to the generation of new hypotheses explaining some aspects of the gene interactions, to be confirmed in wet laboratory experiments. Mining expression data for association rule extraction seems to be more useful to interpret and to understand gene networks: association rules can describe how the expression of one gene may be associated with the expression of a set of genes. It must be noticed that an association rule implies an "association" which is not necessarily a "cause and effect" relationship. Determining the precise nature of the association requires biological knowledge, as emphasized in the preceding paragraph on the mining of chemical reaction databases. This study shows that it becomes possible to develop bioinformatics applications that go further than storing and retrieving expression data, and to propose tools for exploratory data analysis.

5.3 An introduction to Web mining

In the framework of the Semantic Web, the machines are talking to the machines for delivering services to people [18]. To-morrow the Web will be a distributed, shared, declarative and navigable space; it will be mainly exploited by computers solving problems for humans, and providing the results to humans. The semantics of documents on the Web must be accessible to computers. One main element of this semantics is constituted by an explicit model of the domain of data, describing the vocabulary and the structure of informations in relation with the domain of interest. This model must be commonly accepted and shared: this is the essence of the notion of *ontology*, as it is considered in the framework of semantic Web, and for building knowledge systems. For example, let us consider the following list of queries that leads to a series

of different and complex problems:

- *A book on Félix Leclerc.*
- *A book written by Félix Leclerc or a book of Félix Leclerc.*
- *A biography of Félix Leclerc.*
- *An autobiography of Félix Leclerc.*
- *A songbook of Félix Leclerc.*
- *A book on the work of Félix Leclerc.*

For answering these questions, a computer system has to understand the actual meaning of the questions (the "intended meaning" of the user), and the system has to be able to make the difference between "on" in "a book on" and "of" in "a book of", and to understand the difference between terms such as "book", "songbook", "biography", "autobiography"... This is the purpose of ontologies in the framework of Semantic Web and Web mining [41]. Moreover, it can be also very useful for the system to know who is "Félix Leclerc" for answering the above questions (as it should be for a human himself...).

The description of the content of documents may be made explicit by using doculent description languages suc as XML, and a semantics can be attached to documents –and their content– using knowledge representation languages, e.g. description logics, OWL [18]. An intelligent manipulation of documents is based on the exploitation of the content and of the semantics of the documents, with respect to the knowledge on the domain of documents. The technology for the semantic Web is based on the one hand on the use of languages for ontology representation, and for document and resource description such as XML and RDF(S), and on the other hand on the use of intelligent search engines and mining modules for improving the retrieval of adequate resources for problem solving. In this way, information extraction –extraction of key terms from documents– and data mining –especially text mining– may be used for analyzing and classifying documents with respect to their content (the reference [9] may be of interest regarding content-based information retrieval and lattice-based classification of documents).

The *mining of documents on the Web*, or *Web mining*, can be carried out with three main points of view [27, 7]:

- The *mining of the content* of documents, in relation with text mining (see [26] for example).
- The *mining of the structure* of the pages and of the links between pages (hypertext links).
- The *mining of usages* or mining the sets of operations applied on pages.

Web mining can be a major technique in the design of semantic Web: on that base, ontologies can be designed in a semi-automatic way, leading the real-scale ontologies, semi-automatic design rather than manual design of ontologies. Ontologies can be used for *annotating* the documents, and thus to enhance the document mining process, on the base of the content of documents. The Web mining process can be used to improve annotation of documents, and thus the semantics attached to the documents, i.e. content, understandability, and structure.

Moreover, information retrieval can be guided by document mining: key terms are extracted and used to complete domain ontologies, that are in turn used for guiding the data mining process, and so on... Knowledge units extracted from documents can be used for classifying documents according to relations between units and the domain ontology, leading to alternative points of view on documents.

6 Discussion

The KDD process must be carried out in a KDD environment where data mining is guided by domain knowledge, embedded in ontologies and knowledge-based systems. The knowledge units used in knowledge systems may have two major different sources: explicit knowledge that can be given by domain experts, and implicit knowledge that must be extracted from databases of different kinds, e.g. rough data or textual documents. In addition, an important question in the framework of Semantic Web and Web mining for improving the KDD process is to be able to manipulate documents by their content, for searching, for annotating and for classifying the documents. The content-based manipulation of documents allows to solve a number of problems such as information extraction, intelligent information retrieval, content-based document mining... More precisely, the following requirements for knowledge discovery tools are given in [8] :

- The system should represent and present to the user the underlying domain in a natural and appropriate fashion. Objects of the domain should be easily incorporated into queries.
- The domain representation should be extendible by the addition of new concepts or classes formed from queries. These concepts and their representative individuals must be usable in subsequent queries.
- It should be easy to form tentative segmentations of data, to investigate the segments, and to re-segment quickly and easily. There should be a powerful repertoire of viewing and analysis methods, and these methods should be applicable to segments (such as in the Weka system for example [49]).
- Analysts should be supported in recognizing and abstracting common analysis (segmenting and viewing) patterns. These patterns must be easy to apply and modify.

- There should be facilities for monitoring changes in classes or concepts over time.
- The system should increase the transparency of the knowledge discovery process and should document its different stages.
- Analysis tools should take advantage of explicitly represented background knowledge of domain experts, but should also activate the implicit knowledge of experts.
- The system should allow highly flexible processes of knowledge discovery respecting the open and procedural nature of productive human thinking. This means in particular to support the intersubjective communication and argumentation.

The support of knowledge discovery by concept lattices, itemset search and association rule extraction, may be explained as follows [48]. The mathematization of logical structures of concepts and concept hierarchies by formal concepts and concept lattices of formal contexts yields a close relationship between logical and mathematical thinking, which, in particular, allows to activate a rich amount of mathematics to support human reasoning. Especially, the representation of concept lattices by labeled line diagrams enables an interplay between the mathematical analysis of relationships and the logical analysis of data and information, influenced by already existing background knowledge. Therefore, conceptual knowledge discovery, i.e. conceptual information discovery and knowledge creation, can be performed by first looking under the guidance of some purpose for discoveries of information in graphically represented concept lattices, and then creating new knowledge from the discovered information and appropriate pre-knowledge. These two steps should be repeated in a circular process which is open for critic and self-correction.

7 Conclusion

The knowledge discovery in databases process consists in processing a huge volume of data in order to extract knowledge units that can be reused either by an expert of the domain of data or by a knowledge-based system for problem-solving in the domain of data. The KDD process is based on three major steps, data preparation, data mining and interpretation of the extracted units. Moreover, the KDD process is iterative and interactive, and is controlled by an analyst, who is in charge of guiding and validating the extraction process. In addition, the KDD process may take advantage of domain knowledge, i.e. ontologies, knowledge base, for improving the process at every step. Data mining methods are divided into two main categories, symbolic and numerical methods. In this research report, we have mainly focused on symbolic methods, and especially on lattice-based classification, frequent itemset levelwise search, and association rule extraction. These methods are operational and can provide good results in real-world problems. Indeed, three kinds of application have been detailed, an experiment on the mining of chemical reaction databases, an experiment on the mining

of gene expression databases, and finally, a research field with a growing importance, Web mining.

Regarding the future of KDD, there remains many problems to be solved, at every step of the process, especially considering the KDD process as a knowledge-guided process, as we have tried to demonstrate it, and considering the complete environment of a KDD system as a combination of a database and of a knowledge base operations. Another important investigation field for symbolic methods is the extension to the processing of complex data (contrasting with Boolean data). Finally, let us mention that important challenges are linked to the application domains, and must still be undertaken, e.g. biology, chemistry, medicine, space, weather forecast, finance,... At the beginning of this report, we have compared knowledge discovery to gold research or archaeology: first, it is necessary to try to be used with the domain of data, then to apply a number of data mining methods that produce more or less useful results, and then to validate these results. Meanwhile, the analyst has to be patient because the process is iterative –the work may be long without being successful– but it is worth continuing the job, being confident and optimistic!

References

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining Association Rules between Sets of Items in Large Databases. In *Proceedings, ACM SIGMOD Conference on Management of Data, Washington, D.C.*, pages 207–216, 1993.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast Discovery of Association Rules. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 307–328, Menlo Park, California, 1996. AAAI Press / MIT Press.

[3] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th Conference on Very Large Data Bases (VLDB-94)*, pages 478–499, 1994.

[4] M. Barbut and B. Monjardet. *Ordre et classification – Algèbre et combinatoire (2 tomes)*. Hachette, Paris, 1970.

[5] S. Berasaluce, C. Laurenço, A. Napoli, and G. Niel. An Experiment on Knowledge Discovery in Chemical Databases. In J.-F Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, editors, *Knowledge Discovery in Databases: PKDD 2004, Pisa*, Lecture Notes in Artificial Intelligence 3202, pages 39–51. Springer, Berlin, 2004.

[6] S. Berasaluce, C. Laurenço, A. Napoli, and G. Niel. Data mining in reaction databases: extraction of knowledge on chemical functionality transformations. Technical Report A04-R-049, LORIA, Nancy, 2004.

[7] B. Berendt, A. Hotho, and G. Stumme. Towards Semantic Web Mining. In I. Horrocks and J. Hendler, editors, *The Semantic Web - ISWC 2002*, Lecture Notes in Artificial Intelligence 2342, pages 264–278. Springer, Berlin, 2002.

[8] R.J. Brachman and T. Anand. The Process of Knowledge Discovery in Databases. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 37–57, Menlo Park, California, 1996. AAAI Press / MIT Press.

[9] C. Carpineto and G. Romano. *Concept Data Analysis: Theory and Applications*. John Wiley & Sons, Chichester, UK, 2004.

[10] H. Cherfi, A. Napoli, and Y. Toussaint. Toward a text mining methodology using frequent itemset and association rule extraction. In M. Nadif, A. Napoli, E. SanJuan, and A. Sigayret, editors, *Journées de l'informatique Messine (JIM-2003), Knowledge Discovery and Discrete Mathematics, Metz*, pages 285–294. INRIA, 2003.

[11] C. Creighton and S. Hanash. Mining gene expression databases for association rules. *Bioinformatics*, 19(1):79–86, 2003.

[12] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, Cambridge, UK, 1990.

[13] L. Dehaspe and H. Toivonen. Discovery of frequent datalog patterns. *Data Mining and Knowledge Discovery*, 3:7–36, 1999.

[14] M.H. Dunham. *Data Mining – Introductory and Advanced Topics*. Prentice Hall, Upper Saddle River, NJ, 2003.

[15] V. Duquenne. Latticial structures in data analysis. *Theoretical Computer Science*, 217:407–436, 1999.

[16] S. Dzeroski and N. Lavrac, editors. *Relational Data Mining*. Springer, Berlin, 2001.

[17] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. AAAI Press / MIT Press, Menlo Park, California, 1996.

[18] D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster, editors. *Spinning the Semantic Web*. The MIT Press, Cambridge, Massachusetts, 2003.

[19] B. Ganter, P.A. Grigoriev, S.O. Kuznetsov, and M.V. Samokhin. Concept-based data mining with scaled labeled graphs. In K.E. Wolff, H.D. Pfeiffer, and H.S. Delugach, editors, *Conceptual Structures at Work: Proceedings of the 12th International Conference on Conceptual Structures, ICCS 2004, Huntsville, AL*, Lecture Notes in Artificial Intelligence 3127, pages 94–108. Springer, Berlin, 2004.

[20] B. Ganter and S. Rudolph. Formal Concept Analysis Methods for Dynamic Conceptual Graphs. In H.S.Delugach and G. Stumme, editors, *Conceptual Structures: Broadening the Base – 9th International Conference on Conceptual Structures, ICCS-2001, Stanford*, Lecture Notes in Artificial Intelligence 2120, pages 143–156. Springer, Berlin, 2001.

[21] B. Ganter and R. Wille. *Formal Concept Analysis*. Springer, Berlin, 1999.

[22] A. Guénoche. Construction du treillis de galois d'une relation binaire. *Mathématiques, Informatique et Sciences Humaines*, 109:41–53, 1990.

[23] A. Guénoche and I. Van Mechelen. Galois Approach to the Induction of Concepts. In I. Van Mechelen, J. Hampton, R.S. Michalski, and P. Theuns, editors, *Categories and Concepts. Theoretical Views and Inductive Data Analysis*, pages 287–308. Academic Press, London, 1993.

[24] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco, 2001.

[25] D. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. The MIT Press, Cambridge (MA), 2001.

[26] D. Janetzko, H. Cherfi, R. Kennke, A. Napoli, and Y. Toussaint. Knowledge-based selection of association rules for text mining. In R. Lopez de Màntaras and L. Saitta, editors, *16h European Conference on Artificial Intelligence – ECAI'04, Valencia, Spain*, pages 485–489, 2004.

[27] R. Kosala and H. Blockeel. Web Mining: A Survey. *SIGKDD Explorations*, 2(1):1–15, 2000. http://www.acm.org/sigkdd/explorations.

[28] S.O. Kuznetsov. Machine learning and formal concept analysis. In Peter W. Eklund, editor, *Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia*, Lecture Notes in Computer Science 2961, pages 287–312. Springer, 2004.

[29] S.O. Kuznetsov and S.A. Obiedkov. Algorithms for the Construction of Concept Lattices and Their Diagram Graphs. In L. De Raedt and A. Siebes, editors, *Principles of Data Mining and Knowledge Discovery: 5th European Conference, PKDD 2001, Freiburg, Germany*, Lecture Notes in Computer Science 2168, pages 289–300. Springer-Verlag Heidelberg, 2001.

[30] S.O. Kuznetsov and S.A. Obiedkov. Comparing performance of algorithms for generating concept lattices. *Journal of Theoretical Artificial Intelligence*, 14(2/3):189–216, 2002.

[31] N. Lavrac, P.A. Flach, and B. Zupan. Rule Evaluation Measures: A Unifying View. In S. Dzeroski and P.A. Flach, editors, *Inductive Logic Programming, 9th International Workshop, ILP-99, Bled, Slovenia*, Lecture Notes in Computer Science 1634, pages 174–185. Springer, 1999.

[32] H. Mannila. Methods and problems in Data Mining. In F. Afrati and P. Kolaitis, editors, *Database Theory – ICDT'97, 6th International Conference, Delphi, Greece*, Lecture Notes in Artificial Intelligence 1186, pages 41–55. Springer, Berlin, 1997.

[33] H. Mannila, H. Toivonen, and A.I. Verkamo. Efficient algorithms for discovering association rules. In *Proceedings of the 1994 Knowledge Discovery in Databases Workshop, KDD'94*, pages 181–192. AAAI Press, 1994.

[34] R.S. Michalski, I. Bratko, and M. Kubat, editors. *Machine Learning and Data Mining*. John Wiley & Sons LTD, Chichester, 1998.

[35] T.M. Mitchell. *Machine Learning*. McGraw-Hill, Boston, Massachusetts, 1997.

[36] A. Napoli, C. Laurenço, and R. Ducournau. An object-based representation system for organic synthesis planning. *International Journal of Human-Computer Studies*, 41(1/2):5–32, 1994.

[37] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In C. Beeri and P. Buneman, editors, *Database Theory - ICDT'99 Proceedings, 7th International Conference, Jerusalem, Israel*, Lecture Notes in Computer Science 1540, pages 398–416. Springer, 1999.

[38] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Pruning closed itemset lattices for association rules. *International Journal of Information Systems*, 24(1):25–46, 1999.

[39] Z. Pawlak, editor. *Rough Sets. Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht, The Netherlends, 1991.

[40] G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. van de Velde, and B. Wielinga. *Knowledge Engineering and Management: the CommonKADS Methodoloy*. The MIT Press, Cambridge, MA, 1999.

[41] S. Staab and R. Studer, editors. *Handbook on Ontologies*. Springer, Berlin, 2004.

[42] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, and L. Lakhal. Computing iceberg concept lattices with titanic. *Journal of Data and Knowledge Engineering*, 42(2):189–222, 2002.

[43] G. Stumme, R. Wille, and U. Wille. Conceptual Knowledge Discovery in Databases Using Formal Concept Analysis Methods. In J. Zytkow and M. Quafafou, editors, *Principles of Data Mining and Knowledge Discovery (Proceedings PKDD'98, Nantes)*, Lecture Notes in Artificial Intelligence 1510, pages 450–458, Berlin, 1998. Springer.

[44] P.N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proceedings of the 8^{th} ACM International Conference on Knowledge Discovery and Data Mining (KDD'02), Edmonton, Canada*, pages 183–193, 2002.

[45] P. Valtchev, R. Missaoui, and R. Godin. Formal concept analysis for knowledge discovery and data mining: The new challenges. In Peter W. Eklund, editor, *Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia*, Lecture Notes in Computer Science 2961, pages 352–371. Springer, 2004.

[46] P. Vismara and C. Laurenço. An abstract representation for molecular graphs. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 51:343–366, 2000.

[47] J.T.L. Wang, M.J. Zaki, H.T.T. Toivonen, and D. Shasha, editors. *Data Mining in Bioinformatics*. Morgan Kaufmann Publishers, Springer, Berlin, 2004.

[48] R. Wille. Why can concept lattices support knowledge discovery in databases? *Journal of Theoretical Artificial Intelligence*, 14(2/3):81–92, 2002.

[49] I.H. Witten and E. Franck. *Data Mining*. Morgan Kaufmann Publishers, San Francisco, California, 2000. (Practical machine learning tools and techniques with Java implementations – Weka).

[50] M.J. Zaki and C.-J. Hsiao. Charm: An efficient algorithm for closed itemset mining. In R. Grossman, J. Han, V. Kumar, H. Mannila, and R. Motwani, editors, *Second SIAM International Conference on Data Mining, Arlington*, 2002.