Formal Concept Analysis for Digital Ecosystem

Huaiguo Fu
Telecommunications Software & Systems Group
Waterford Institute of Technology, Waterford, Ireland
hfu@tssg.org

Abstract

Formal Concept Analysis (FCA) is an effective tool for data analysis and knowledge discovery. Concept lattice, which is derived from mathematical order theory and lattice theory, is the core of FCA. Many research works of various areas show that concept lattices structures is an effective platform for data mining, machine learning, information retrieval, software engineer, etc. This paper offers a brief overview of FCA and proposes to apply FCA as a tool for analysis and visualization of data in Digital ecosystem, and also discusses the applications of data mining for Digital ecosystem.

1 Introduction

Concept is the base of human's thinking and logic. We can distinguish and understand the various different objects of the real world by the concepts that describe the categories and attributes of the objects. Concept is also important for data analysis in computer science. Formal Concept Analysis (FCA) is a method for data analysis, information management and knowledge representation that takes advantage of the features of formal concepts.

The core of FCA is concept lattice. Theoretical foundation of concept lattice founds on the mathematical lattice theory [3, 11]. Lattice is a popular mathematical structure for modeling conceptual hierarchies. Concept lattice is a method for deriving conceptual structures out of data. For concept lattice, we study the relations between objects and attributes in a formal context, and how objects can be hierarchically grouped together according to their common attributes. Certain object subset and the set of their common attributes can represent each other, such duality sets form a formal concept, which the attribute subset is called intent and the object subset is called extent. Among the formal concepts, it exists an order relation, they form a complete lattice: concept lattice. Each node in the lattice is a concept and the corresponding graph (Hasse diagram) is considered as the generalization and specialization relationships between concepts. Such graphical structure represents directly and visually the relations of conceptual hierarchies. It allows us to analyze and mine the complex data for such as classification, association rules mining, clustering, etc.

The application of concept lattice has been an area of active and promising research in various fields such as knowledge discovery, information retrieval, software engineer and machine learning. For example, in the context of association rule mining [1], which consists in finding all associations and correlation among data items with a certain criterion (degree of support and confidence), it's a hard problem to find all frequent sets in a large data. The frequent sets generation is the most important step for association rules. Concept lattice structure [13] has shown to be an effective tool for finding frequent concepts for association rules [17, 20, 18, 24]. The problem of finding frequent sets from data for association rules can be reduced to find frequent concepts with concept lattice. And it's possible to prune the number of rules produced without information loss using closed set lattice [16].

We propose to use FCA as a tool for data analysis, information management and knowledge representation in Digital ecosystem.

Digital ecosystem (DE) is a new concept. In the EU research community, the Digital Ecosystem aims at providing to small and micro enterprises (SMMEs) ICT applications and services which improve their efficiency, business integration and synergies within EU territories, but also enabling their integration of local value chains within the global market. In the natural world, an ecosystem is a system whose members benefit from each other's participation via symbiotic relationships. For the digital ecosystem, it is a "digital environment" populated by "digital species" or "digital components" which can be software components, applications, services, knowledge, business processes and models, training modules, contractual frameworks, law, etc. A digital component is any useful idea, expressed by a language (formal or natural), digitalised and transported within the ecosystem, and which can be processed by humans or by

Proceedings of the 5th International Conference on Machine Learning and Applications (ICMLA'06)
0-7695-2735-3/06 $20.00 © 2006 IEEE

computers. The digital ecosystem infrastructure supports the description, compositions, evolution, integration, sharing and distribution of the digital components and of knowledge.

The digital ecosystem initiative aims at fostering a cultural change in enterprise networking and in business practices. It innovates and impacts on three aspects: technology, business practices and knowledge.

In the digital ecosystem, some knowledge is existing, but some knowledge is previously unknown, implicit, hidden in large data. So we should extract the knowledge from large data. The techniques of data mining (DM) are widely used in research and application to look for relationships and knowledge that are implicit in large volumes of data and are interesting in the sense of impacting an organization's practice. Hence, we propose to use the techniques of data mining to extract knowledge for the digital ecosystem. We will discuss main issues of application of data mining for the digital ecosystem.

Contributions. This paper makes the following contributions: propose to use the technologies of data mining to extract knowledge for digital ecosystem and propose and describe a new tool, formal concept analysis, for digital ecosystem to analyze and represent data and knowledge.

2 Knowledge extraction in DE

Digital Ecosystems are emerging as a novel approach for the catalysis of sustainable development driven by networks of micro- and small enterprises, enabled by ICT services and intelligent cooperative solutions, linking multitudes of socio-economic actors and ICT solutions that are affordable, trustworthy, adaptive and evolutionary. Dynamic and remote collaboration and interaction in structured and unstructured environments are catalysed by new approaches and ICT technologies, which consider the knowledge, ICT solutions and services as digital ecosystem entities which exhibit the behavior of natural organisms. Thus, the infrastructure and the services supporting organisational interaction and networking will form a digital ecosystem considered a pervasive common infrastructure carrying knowledge, models and services, where complex heterogeneous, human and digital entities and systems are themselves composed of simpler subsystems.

In order to extract the implicit knowledge from the digital ecosystem, and help companies to provide better, customized services and support decision making. We discuss the main issues of application of data mining for the digital ecosystem in this section.

2.1 Main aims of DM for DE

For the digital ecosystem, computer science research focuses on the seamless management of distributed, multicentric and pervasive ICT networks carrying services and representations of knowledge, enabling the creation of a self-organising environment that supports the continuous evolution of business models and software services, by exploiting paradigms from biology and economics.

The application of the data mining techniques should consider the features of the digital ecosystem. The purpose of the applications of DM is to develop efficient DM algorithms that scale up large distributed data sets, integrate efficient DM algorithms and techniques, and P2P distributed computing environment for extracting knowledge from large amounts of large and complex distributed data for DE. The main research works will focus on :

- Robustness, reliability, sustainability, and scalability of distributed data mining techniques for DE
- Automatic, autonomic and dynamic DM processes for DE
- Dynamic Peer-to-peer (or abbreviated P2P) architectures for heterogeneous distributed and complex data
- Seamless interoperability with service oriented platforms
- Recursive, reflexive, and self-reinforcing knowledge discovery

We should provide dynamic, scalable and flexible DM algorithms for extracting knowledge efficiently from the heterogeneous, high dimensional and distributed data. We can use various specific algorithms or approaches for the same task, because many algorithms or approaches may exert efficient performance on specific data (particular in size, density, and type etc.) and distributed environment. The system should analyze the characteristics of the data, and then automatically choose a befitting algorithm. Different sites may perform different algorithms or approaches for the same task depending on different characteristics of each site. For example, there are lots of clustering algorithms, but most of them is only suitable for one type of data. Maybe many different types of data in the same or different sites for distributed data. Hence each site can use one suitable clustering algorithm or combination of several clustering algorithms.

According to the application, we can unify some data mining tasks. For example, if we need both clustering and association rule mining for huge distributed transaction data, we can use some techniques to unify these two tasks for avoiding large amounts of repetitious computing.

2.2 Challenges of DM for DE

In recent years, DM has attracted a lot of attention among the fields of research and applications. Many techniques and systems of DM have been proposed. However, the data and infrastructure of DE are very complex. There are many challenges of the applications of DM for DE such as heterogeneous data, complex data, security, privacy and autonomy of local databases, network topology and transmission scheme. We need to develop more scalable and more efficient techniques of DM for DE.

Data mining and knowledge discovery can benefit from the use of distributed data mining (DDM) techniques to improve mining performance of huge data or distributed data. Although there are many efficient algorithms and techniques for mining centralized data sets, it's inefficient or incapable to deal with huge data sets or distributed data sets.

There are two main reasons to choose DDM. The first one is that data is very large. If data is too large, it's hard to store it at a single site, or it's inefficient or incapable to mine such large data at a single site. In such a case, data may be decomposed into some parts that are distributed at different sites. Then we perform the data mining operations for each site. At the end, the mining results of each site are combined to gain global results. This will optimize centralized data mining since the work load is distributed among the sites.

The second reason is that we need to deal with inherent distributed data sets. In fact, various wired and wireless networks such as internet, intranets, local area networks, ad hoc wireless networks and sensor networks etc. produce many distributed resources of data. These distributed data need to be mined to gain global patterns, models or knowledge. The straightforward solution is to transfer all data to a central site, where data mining is done. However, even if we have enough capacity to handle the data storage and data mining at a central site, it may be too expensive to transfer the local data sets to the central site. On the other hand, the privacy issue is playing an important role in the emerging distributed data. The distributed data sets may not be transferred because of privacy, security or autonomy of the data sets. Therefore, DDM is an effective and scalable solution for mining huge and distributed data sets in distributed computing environments.

2.3 Complex data in DE

There are large amounts of distributed data in DE. Most of distributed data is heterogeneous, complex and noisy. It's hard to deal with heterogeneous and complex data. Distributed data can be divided into two categories: homogeneous and heterogeneous. In homogeneous data, the databases located at different sites have the same attributes and in the same format, while in heterogeneous data, the attributes at each site are different or in different format. Heterogeneous data is more complex than homogeneous data for DDM tasks.

Most studies on DDM assume that local databases are homogeneous. So many DDM algorithms only deal with homogeneous data. If the local databases are heterogeneous, we need to adopt different techniques to deal with them. Integrating local models of heterogeneous data is hard for many data mining tasks. Therefore, developing DDM algorithms that can handle heterogeneous data is becoming increasingly important.

Many real data are high dimensional, high dense, non static, unbalanced. Increasingly complex data sources, structures, and types (like natural language text, images, time series, continuous data streams, multi-relational and object data types etc.) are emerging. It requires the development of new methodologies, algorithms, tools, and services to mine such complex data. One solution for managing the complex data for DDM is to unify different data. For example, we can use XML to present complex data.

Sometimes, complexity of data rests with noise in the data. Real world data is dirty and noisy. In a large database, many of the attribute values will be inexact or incorrect, or there are some missing attributes and missing attribute values. Data noise may affect DDM results, so high quality data for DDM is needed. One solution is data preprocessing such as data cleaning, data transformation, data reduction.

2.4 Complex infrastructures in DE

Distributed environment is the base of DE. DDM needs effective infrastructures for distributed large-scale and high-performance computing and data processing. Various wired and wireless networks offer the distributed computing environment. Recently, P2P or grid is considered as more and more important distributed computing environment in DDM.

In centralized data mining, the main concern for the efficiency of a data mining algorithm is its I/O and/or CPU time. In a distributed environment, the communication cost should be considered, it may be a bottleneck in DDM [2, 19]. The cost of transferring large blocks of data may be prohibitive and result in very inefficient implementations in DDM. For a slow network, the communication cost will dominate the overall cost. The communication cost is determined by the infrastructures of the distributed environment, the network bandwidth and the number of messages that are sent across the network. In order to reduce the communication cost, many DDM methods are used to minimize the number of messages sent. Some methods also attempt to load-balance across sites to prevent performance from being dominated by the time and space usage of any individ-

ual site. We consider that one important method is to choose a suitable distributed infrastructures and computing service. The distributed computation infrastructure of P2P or grid is very suitable for DDM. P2P or Grid can provide an effective computational support for DDM applications.

A grid is a geographically distributed computation infrastructure composed of a set of heterogeneous machines that users can access via a single interface. A grid environment provides high performance computing facilities and transparent access to them in spite of their remote location, different administrative domains and hardware and software heterogeneous characteristics. Grid computing provides a novel distributed environment, computational model, and unprecedented opportunities for unlimited computing and storage resources. It's distinguished from conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and, in some cases, high-performance orientation. Grids can be used as effective infrastructures for distributed high-performance computing and data processing [7].

DDM on grid, although is a fairly new research topic, has been very active in data mining community. The main disadvantage of grid is that grid software and standards are still evolving. The development of DDM on grid isn't easy.

P2P architecture is a type of network in which each workstation has equivalent capabilities and responsibilities. This differs from client/server architectures. Generally, P2P networks are used for sharing files, but a P2P network can also mean Grid Computing. Techniques and applications of P2P for DDM can be found in [23].

The primary disadvantage of P2P is the tendency of computers at the edge of the network to fade in and out of availability. Also, accountability for the actions of network participants could be a difficult problem. Several high-profile implementations have shown that architecture, security, and systems management issues are difficult to control. For these reasons, system managers often prefer to operate P2P systems as separate isolated entities. But, doing so is often impossible for practical applications.

3 Concept lattice

Concept lattice [11] and Closed itemset lattice are based on order theory and lattice theory [4, 22]. They are used to represent the order relation of concepts or closed itemsets. Concept lattice describes the character of the set pair: intent and extent of concept. Closed itemset lattice emphasizes the representation of the character of itemset.

In this section, we define some basic notions: Data context, Closure operator, Formal concept, Concept lattice, Closed itemset and Closed itemset lattice.

Definition 3.1 Data context *is defined by a triple $(G; M; R)$, where G and M are two sets, and R is a relation between G and M. The elements of G are called objects or transactions, while the elements of M are called attributes or items.*

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
1	×	×					×	
2	×	×					×	×
3	×	×	×				×	×
4	×		×				×	×
5	×	×		×		×		
6	×	×	×	×		×		
7	×		×	×	×			
8	×		×	×		×		

Figure 1. An example of data context

A data context is usually represented by the binary data, but in practice, the values of attribute are not binary, we can transform many-valued data context to binary values context by concept scaling [11].

Definition 3.2 *Given a subset $A \subseteq G$ of objects from a data context $(G; M; R)$, we define an operator that produces the set A' of their common attributes for every set $A \subseteq G$ of objects to know which attributes from M are common to all these objects:*

$A' := \{m \in M \mid gRm \text{ for all } g \in A\}$.

Dually, we define B' for subset of attributes $B \subseteq M$, B' denotes the set consisting of those objects in G that have all the attributes from B:

$B' := \{g \in G \mid gRm \text{ for all } m \in B\}$.

These two operators are called the **Galois connection** *for $(G; M; R)$. These operators are used to determine a formal concept.*

So if B is an attribute subset, then B' is an object subset, and then $(B')'$ is an attribute subset. We have:
$B \subseteq M \Rightarrow B'' \subseteq M$. Correspondingly, for object subset A, we have: $A \subseteq G \Rightarrow A'' \subseteq G$.

Thus we define two **closure operators** *as $B \rightarrow B''$ for set M and $A \rightarrow A''$ for set G.*

For example, Figure 1 represents a data context. $G(1, 2, 3, 4, 5, 6, 7, 8)$ is the set of objects, and $M(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8)$ is the set of items. The crosses in the table describe the relation R of G and M.

Definition 3.3 *A* **formal concept** *of the data context (G, M, R) is a pair (A, B) with $A \subseteq G$, $B \subseteq M, A = B'$ and $B = A'$. A is called extent, B is called intent.*

Definition 3.4 *If (A_1, B_1) and (A_2, B_2) are concepts, $A_1 \subseteq A_2$ (or $B_2 \subseteq B_1$), then we say that there is a hierarchical order between (A_1, B_1) and (A_2, B_2).*

All concepts with the hierarchical order of concepts form a complete lattice called **concept lattice**.

For example, (68, $a_1a_3a_4a_6$) is a concept of the data context of Figure 1. $a_1a_3a_4a_6$ is intent of (68, $a_1a_3a_4a_6$), and 68 is extent of (68, $a_1a_3a_4a_6$).

Definition 3.5 *An itemset $C \subseteq M$ is a* **closed itemset** *iff $C'' = C$.*

Thus, a closed itemset is intent of a formal concept. This definition is very important for closed itemset algorithm.

A formal concept or closed itemset describes more a stricter relation between objects and attributes than itemset of association rules mining. For itemset, one attribute set maps to an object set, it's injective; but for formal concept, there is a bijection between one attribute set and one object set. The intent of a concept is a closed itemset and it's a maximal itemset.

For example, $\{a_1,a_7\}$ is a closed itemset of the data context of Figure 1.

Definition 3.6 *If C_1 and C_2 are closed itemsets, $C_1 \subseteq C_2$, then we say that there is a hierarchical order between C_1 and C_2.*

All closed itemsets with the hierarchical order of closed itemsets form of a complete lattice called **closed itemset lattice**.

4 Analysis and visualization of data with FCA in DE

Formal Concept Analysis provides a natural platform for data analysis and knowledge representation. FCA is different from some of the traditional, statistical means of data analysis and knowledge representation because of its focus on human-centered approaches. Formal concept possesses the same features as philosophical concept. From the formal concepts, we can analyze data such as revealing stronger association or relation between itemset and the set of their common objects, classifying objects, generating implications of attributes or knowledge rules, extracting the hierarchical relation among formal concepts, etc.

FCA also provides an effective tool of knowledge visualization. Concept lattice can show how objects can be hierarchically grouped together according to their common attributes, and the relations between the formal concepts. Concept lattice facilitate discussion and exploration of conceptual structures. FCA has been examined with respect to principles of knowledge representation. Wille [21] identifies ten functions of knowledge processing (exploring, searching, recognizing, identifying, analyzing, investigating, deciding, improving, restructuring and memorizing) and investigates how these are supported by FCA.

Several algorithms were proposed to generate concepts or concept lattices on a data context, for example: Bordat [5], Ganter (NextClosure algorithm) [10], Chein [6], Norris [14], Godin [12] and Nourine [15], etc. We can use the formal concepts or concept lattices to analyze and represent the data. Concept lattice can be also applied to distributed data to analyze and represent knowledge in DE [8, 9].

For example, using lattice algorithms, we can generate concepts or concept lattices on a data context (see Figure 1). All formal concepts and the concept lattices on the data context (see Figure 1) are shown in Figure 3. The closed itemset lattice of the data context of Figure 1 is presented in Figure 4. The Figure 2 shows intents of the data context (see Figure 1).

No.	Intent	No.	Intent
1	$\{a_1\}$	10	$\{a_1a_7a_8\}$
2	$\{a_1a_2\}$	11	$\{a_1a_2a_4a_6\}$
3	$\{a_1a_3\}$	12	$\{a_1a_2a_7a_8\}$
4	$\{a_1a_4\}$	13	$\{a_1a_3a_4a_5\}$
5	$\{a_1a_7\}$	14	$\{a_1a_2a_3a_4a_6\}$
6	$\{a_1a_2a_3\}$	15	$\{a_1a_3a_7a_8\}$
7	$\{a_1a_2a_7\}$	16	$\{a_1a_2a_3a_4a_6\}$
8	$\{a_1a_3a_4\}$	17	$\{a_1a_2a_3a_7a_8\}$
9	$\{a_1a_4a_6\}$	18	$\{a_1a_2a_3 \ldots a_7a_8\}$

Figure 2. All intents of the data context (see figure 1)

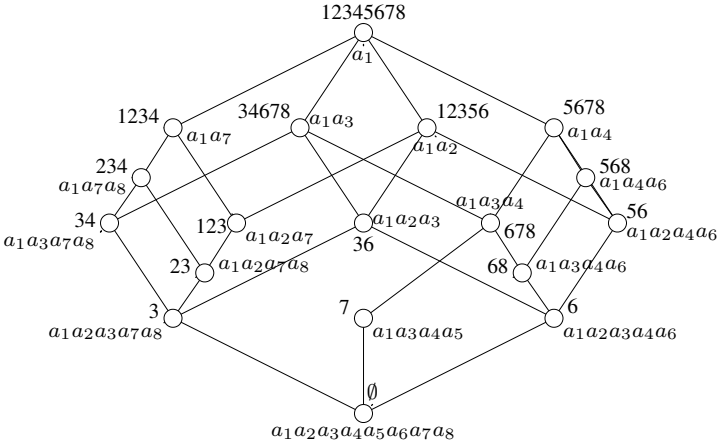

Figure 3. An example of concept lattice

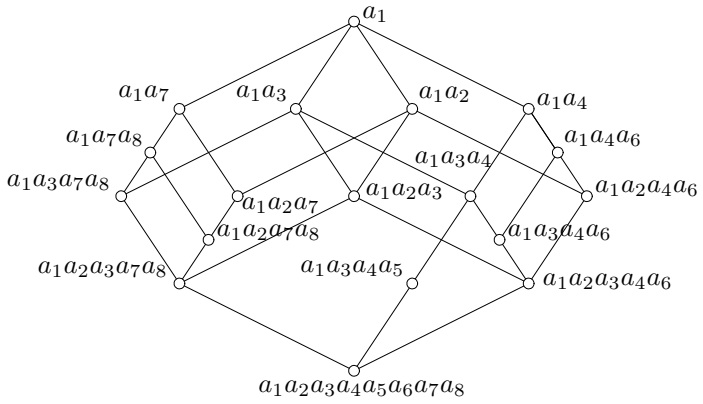

Figure 4. An example of closed itemset lattice

5 Conclusion

Formal Concept Analysis provides a natural effective platform for data analysis and knowledge representation. In this paper, we propose FCA as a tool of data analysis and representation for digital ecosystem. We also propose to use the technologies of data mining to extract knowledge from huge and complex heterogeneous distributed data in the DE. Furthermore, some issues of the applications of data mining for Digital ecosystem is discussed.

Acknowledgements

This work is supported by the project of EU IST Network of Excellence "OPAALS".

References

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. Verkamo. *Advances in Knowledge discovery and Data Mining*, chapter Fast discovery of association rules, pages 307–328. AAAI/MIT Press, 1996.

[2] P. B. Bhat, C. S. Raghavendra, and V. K. Prasanna. Efficient collective communication in distributed heterogeneous systems. *Journal of Parallel and Distributed Computing*, 63(3):251–263, 2003.

[3] G. Birkhoff. *Lattice Theory*. American Mathematical Society, Providence, RI, 3rd edition, 1967.

[4] G. Birkhoff. *Lattice theory (third ed.)*. Number XXV in American Mathematical Society Colloquium Publication. American Mathematical Society, 1973.

[5] J. Bordat. Calcul pratique du treillis de galois d'une correspondance. *Mathématiques, Informatiques et Sciences Humaines*, 24(94):31–47, 1986.

[6] M. Chein. Algorithme de recherche des sous matrices premières d'une matrice. *Bull.Math.R.S.*, 13, 1969.

[7] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke. The Data Grid: Towards an Architecture For the Distributed Management and Analysis of Large Scientific Datasets, 1999.

[8] H. Fu and E. Mephu Nguifo. Partitioning large data to scale up lattice-based algorithm. In *Proceedings of ICTAI03*, pages 537–544, Sacramento, CA, November 2003. IEEE Computer Press.

[9] H. Fu and E. Mephu Nguifo. Mining frequent closed itemsets for large data. In *Proceedings of The 2004 International Conference on Machine Learning and Applications (ICMLA04)*, Louisville, USA, December 2004.

[10] B. Ganter. Two basic algorithms in concept analysis. Technical report, Darmstadt University, 1984.

[11] B. Ganter and R. Wille. *Formal Concept Analysis. Mathematical Foundations*. Springer, 1999.

[12] R. Godin, G. Mineau, and et al. Méthodes de classification conceptuelle basés sur les treillis de galois et application. *Revue d'intelligence artificielle*, pages 105–137, 1995.

[13] E. Mephu Nguifo, M. Liquiere, and V. Duquenne. *Journal of Experimental and Theoretical Artificial Intelligence (JETAI) Special Issue on Concept Lattice-based theory, methods and tools for Knowledge Discovery in Databases*. Taylor and Francis, 2002.

[14] E. Norris. An algorithm for computing the maximal rectangles in a binary relation. *Revue Roumaine Math. Pures et Appl.*, XXIII(2):243–250, 1978.

[15] L. Nourine and O. Raynaud. A fast algorithm for building lattices. *Information Processing Letters*, 71:199–204, 1999.

[16] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. *Lecture Notes in Computer Science*, 1540:398–416, 1999.

[17] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemsets lattices. *Journal of Information Systems*, 24(1):25–46, 1999.

[18] J. Pei, J. Han, and R. Mao. CLOSET: An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30, 2000.

[19] F. Provost. Distributed Data Mining: Scaling Up and Beyond. In H. Kargupta and P. Chan, editors, *Advances in Distributed Data Mining*. MIT/AAAI Press, 2000.

[20] P. Valtchev, R. Missaoui, and et al. Generating frequent itemsets: two novel approaches based on galois lattice theory. *Journal of Experimental and Theoretical Artificial Intelligence (JETAI) Special Issue on Concept Lattice-based theory, methods and tools for Knowledge Discovery in Databases*, April 2002.

[21] R. Wille. Conceptual landscapes of knowledge: a pragmatic paradigm for knowledge processing.

[22] R. Wille. Restructuring Lattice Theory. In I. Rival, editor, *Symposium on Ordered Sets*, pages 445–470. University of Calgary, Boston, 1982.

[23] R. Wolff and A. Schuster. Association Rule Mining in Peer-to-Peer Systems . In *Third IEEE International Conference on Data Mining*, Melbourne, FL, November 2003.

[24] M. J. Zaki and C.-J. Hsiao. CHARM: An efficient algorithm for closed itemset mining. Technical Report 99-10, Rensselaer Polytechnic Institute, 1999.