Fall 12-2022

LEARNFCA: A FUZZY FCA AND PROBABILITY BASED APPROACH FOR LEARNING AND CLASSIFICATION

Suraj Ketan Samal
University of Nebraska - Lincoln, sksamal@gmail.com

LEARNFCA: A FUZZY FCA AND PROBABILITY BASED APPROACH FOR LEARNING AND CLASSIFICATION

by

Suraj Ketan Samal

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor Jitender Deogun

Lincoln, Nebraska

December, 2022

LEARNFCA: A FUZZY FCA AND PROBABILITY BASED APPROACH FOR LEARNING AND CLASSIFICATION

Suraj Ketan Samal, M.S.

University of Nebraska, 2022

Adviser: Jitender Deogun

Formal concept analysis(FCA) is a mathematical theory based on lattice and order theory used for data analysis and knowledge representation. Over the past several years, many of its extensions have been proposed and applied in several domains including data mining, machine learning, knowledge management, semantic web, software development, chemistry ,biology, medicine, data analytics, biology and ontology engineering.

This thesis reviews the state-of-the-art of theory of Formal Concept Analysis(FCA) and its various extensions that have been developed and well-studied in the past several years. We discuss their historical roots, reproduce the original definitions and derivations with illustrative examples. Further, we provide a literature review of it's applications and various approaches adopted by researchers in the areas of data-analysis, knowledge management with emphasis to data-learning and classification problems.

We propose LearnFCA, a novel approach based on FuzzyFCA and probability theory for learning and classification problems. LearnFCA uses an enhanced version of FuzzyLattice which has been developed to store class labels and probability vectors and has the capability to be used for classifying instances with encoded and unlabelled features. We evaluate LearnFCA on encodings from three datasets - mnist, omniglot and cancer images with interesting results and varying degrees of success.

ACKNOWLEDGMENTS

First and foremost, I thank my advisor, Dr Jitender Deogun. Dr Deogun is always very kind hearted and I am grateful for his valuable guidance over many years, his generosity with time, and all meetings we have had over the course of last few years, both for my professional and personal growth. He has mentored me with variety of things, including my course assignments, research topics and brings a great deal of excitement to all the research areas he works with. I am grateful for all the members of my thesis committee, including Dr Variyam and Dr Hasan for advising me with research directions, working with me for research problems during the entire course of my program. Dr Variyam provided me with valuable support, interesting research directions and pushed my thinking abilities. Dr Hasan mentored me closely with the classes he taught, giving me opportunity for some valuable teaching experience that I will cherish my entire career.

I owe a great deal to Adam Voshall, whose vast experience, and expertise in computer science and biological sciences has led to development of most of the work in this thesis. Many colleagues and members, past and present, have contributed to this work, either through collaboration, reviewing my writings, working on code together or providing valuable suggestions during discussions. I would like to thank to all the members of our group especially Mohammed Alali, Aziza Alzadali, Sairam Behera, Baset Hamza and Saravanan Raju for constant encouragement and invaluable support.

This thesis is the result of research work done as a member of Computer Science Department at the University of Nebraska-Lincoln. I am thankful to the CSE department for partially supporting this research. I also thank TRIO Programs at UNL for the assistantship support during the period of writing this thesis.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Formal Concept Analysis

Formal Concept Analysis (FCA) was introduced in the early 1980's by Rudolf Wille as a mathematical theory of complete lattices. It builds on original theory of ordered sets and lattices developed by Garrett Birkhoff and others in the 1930's. It relies on a basic notion of concept similar to a traditional approach to concepts as in traditional logic. It is a formal way of deriving a concept hierarchy from a collection of objects and their properties.

Development of Formal concept analysis(FCA) became a well-known technique that provided real-world meaning to the mathematical order theory. This led to transformation of two dimensional data tables into algebraic structures or complete lattices, which could then be used for better visualization and interpretation of data. A data item generally represents a heterogeneous relation between object and a set of attributes, tabulating pairs of the form "object g has attribute m". A data table consists of many such data items, which when transformed is referred to as a *formal context*.

In this theory, a *formal concept* is defined to be a pair (A, B), where A is a set of objects (called the extent) and B is a set of attributes (the intent) such that

- the extent A includes all objects that have the attributes in B, and
- the intent B includes all attributes shared by the objects in A.

The formal concepts of a given formal context can be ordered in a hierarchy called a ”concept lattice.” Intuitively, a concept in the hierarchy is a representation of set of objects that share the same properties. Concepts at different level in the hierarchy are known as sub-concepts or super-concepts. A sub-concept represents a subset of the objects and a super-set of the properties in the concepts above it. The concept lattice enables a graphical visualization in terms of a ”line diagram”, which becomes very useful for providing meaning to the data. The lattice is then used to explore concepts, associations and implication rules which are more meaningful and then can be used for generating recommendations. Sometimes, the lattices can get too large for visualization, however many recent techniques like incremental algorithms, decomposing lattices into smaller sub-lattices, ice-berging have been proven useful for easier interpretation and made FCA more practical to be used in big-data applications.

The mathematical theory of Formal Concept Analysis(FCA) has been widely used since its inception to study various characteristics of data and organizing its various components, exploring relationships, mining implication rules, extracting hidden knowledge, creating learning models and even clustering and classifying data. FCA has been used in various domains including data mining, text mining, machine learning, knowledge management, semantic web, software development, chemistry ,biology, medicine, data analytics, biology and ontology engineering with varying degree of success. Various extensions of FCA have been proposed by researchers, which include Fuzzy FCA, Rough FCA, Monotone FCA, Temporal FCA, Temporal Fuzzy FCA to meet specific purposes.

We refer the reader to [4] for a comprehensive survey of state-of-art of FCA and lattice based algorithms. Introduction and applications of FCA are reviewed thoroughly in [5], and basic mathematical foundations of FCA are covered in [6]. FCA forms an integral part of three international conferences, namely, ICFCA (International Conference on Formal Concept Analysis), CLA (Concept Lattices and Their Applications), and ICCS (International Conference on Conceptual Structures).

Definition 1 *A Formal Context is a triple $K = (G, M, I)$, where G is a set of objects, M is a set of attributes, and I is a binary relation from G to M, where each pair $(g, m) \in I$ has a membership value in $\{0, 1\}$. When $(gIm) = 0$, object g has the attribute m and when $(gIm) = 1$, object g doesnot have attribute m.*

A FCA context is a triple of sets $K = (G, M, I)$, where $I \subseteq G \times M$ is a binary relation. These sets are generally represented using a table consisting of set of rows G (called objects), columns M (called attributes) and crosses representing the incidence relation I. Table 1.1 shows an example of such a context where objects are some undergraduate majors, attributes are various undergraduate courses offered and the incidence relation shows the pre-requisite requirements for each undergraduate major. A student of ”Computer Engineering” major needs to take pre-requisite courses namely, ”Computer Architecture”, ”Digital Signal Processing”, ”Intro to Stats” and ”Calculus II”. When such a formal context is given, concepts are derived and ordered using a subconcept-superconcept relation. These concepts are then arranged in a lattice or more commonly called a line diagram which is easier to visualize.

Describing formally, for a set of objects $O \subseteq G$, the set of common attributes can be defined by:

$A = O' = \{m \in M | (o, m) \in I \text{ for all } o \in O\}$

Take the attributes that describe the major ”Computer Engg” in Table 1.1, for instance. By collecting all required objects of this context that share these attributes,

	Algorithm Analysis	**Software Design**	**Computer Architecture**	**Digital Signal Processing**	**Programming Concepts**	**Discrete Math**	**Cryptography**	**Systems Biology**	**Intro to Stats**	**Calculus II**
Computer Science	X		X		X	X	X			
Software Engg	X	X			X	X			X	
Computational Biology	X				X	X		X		
Computer Engg			X	X					X	X
Electrical Engg			X	X					X	X
Mathematics						X	X		X	X

Table 1.1: A formal context of undergraduate majors and pre-requisite courses at UNL

we get to a set O which consists of ”Computer Engg” and ”Electrical Engg”. This set O of objects is related to the set A consisting of the attributes “Computer Architecture”, “Digital Signal Processing” and “Intro to Stats” and ”Calculus II”.

$O = A' = \{o \in G | (o, m) \in I \text{ for all } m \in A\}$

That is, O is the set of all objects sharing all attributes of A, and A is the set of all attributes that are valid descriptions for all the objects contained in O. Each such pair (O, A) is called a formal concept(or concept) of the given context. The set $A = O'$ is called the intent, while $O = A'$ is called the extent of the concept (O, A).

Definition 2 *A Formal Concept C of a formal context $K = (G, M, I)$ is a pair $C = (O, A)$, where, for $O \subseteq G$, $A \subseteq M$, $A' = O$ and $O' = A$.*

There is a natural hierarchical ordering relation between the concepts of a given context that is called the subconcept-superconcept relation.

$(O_1, A_1) \leq (O_2, A_2) \Leftrightarrow (O_1 \subseteq O_2 \Leftrightarrow A_2 \subseteq A1)$

A concept $C_1 = (O_1, A_1)$ is called a sub-concept of a concept $C_2 = (O_2, A_2)$ (or equivalently, C_2 is called a super-concept of a concept C_1) if the extent of C_1 is a subset of the extent) of C_2 (or equivalently, if the intent of C_1 is a super-set of the intent of C_2). For example, the concept with intent "Computer Architecture" is a sub-concept of a concept with intent "Computer Architecture", "Digital Signal Processing", "Intro to Stats" and "Calculus II." With reference to Table 1.1, the extent of the latter is composed of majors "Computer Engg" and "Electrical Engg", while the extent of the former is composed of "Computer Engg", "Electrical Engg" and "Computer Science".

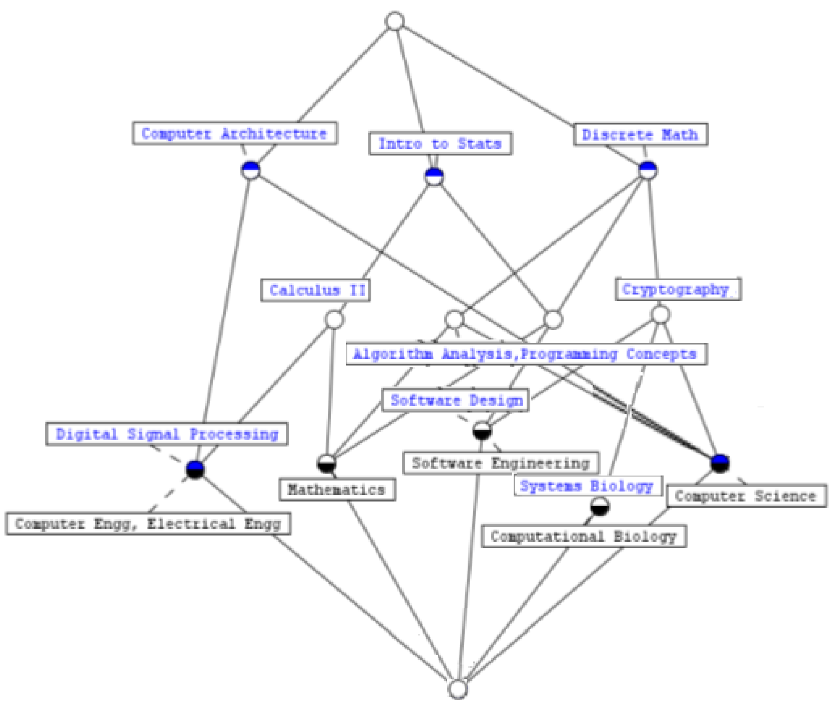

Figure 1.1: Concept Lattice Diagram of the Formal Context in Table 1

The set of all concepts of a formal context ordered by this subconcept-superconcept relation can be organized into a lattice called the *concept lattice* of the context, i.e.

every subset of concepts has a *meet* and *join* w.r.t. $\leq$. Since, concept lattices are ordered sets, they can be visualized by line diagrams, with nodes representing concepts and edges connecting a pair of neighboring nodes w.r.t $\leq$. Figure 1.1, shows the line diagram of the concept lattice of the formal context shown in Table 1.1. The circles represent the formal concepts, boxes with text in *black* (upwards) represent the objects used to name the concept and boxes in *blue* (downwards) represent the attributes used to name the concept. To retrieve the extent of a formal concept, one has to collect all objects on all paths leading down from the corresponding node. Similarly, to retrieve the intent of a formal concept, all paths leading up from the corresponding node has to be traced for collecting all attributes. The top and bottom concepts in the lattice are special called the *infimum* and *supremum* nodes. The top concept contains all objects in its extent. The bottom concept contains all attributes in its intent. A concept is a sub-concept of all concepts that can be reached by travelling upwards. This concept will inherit all attributes associated with these super-concepts.

We can derive all the information in the formal context of Table 1.1 by traversing through the line diagram and using the subconcept-superconcept relationship. For example, the major ”Computer Science” is described by the attributes “Algorithm Analysis”, “Computer Architecture”, “Programming Concepts”, ”Discrete Math” and “Cryptography.” An interesting aspect of FCA which has many applications is the concept of *attribute implication*. For subsets $A, B \subseteq M$, one has $A \to B$ if $A' \subseteq B'$. A set of *implication* rules can be reduced to a minimal subset called an implication base. Some examples include Duquenne-Guigues base (Guigues, Duquenne 1986), which is cardinality minimal, and proper premise base [6]. Ganter & Wille [6] described a knowledge discovery procedure called Attribute Exploration which along with inputs from domain expert can be used to derive a complete useful implication base from the lattice. In many cases, actual data is in the form of many-valued contexts, which can be reduced to binary contexts by a method known as conceptual scaling, first described by Ganter & Wille [6].

1.2 Motivation

The area of Formal Concept Analysis(FCA) has seen a huge growth in the last 40 years. Many generalizations of FCA have been proposed and applied to applications in various domains. Use of FCA in data mining hasn't been investigated thoroughly until recent 2010's [4, 5]. Applications of generalizations of FCA like Fuzzy FCA, Monotone FCA and Temporal FCA in data analysis still needs a lot of research. The role of FCA in machine learning also needs a great deal of exploration. Interesting areas of research include integration of FCA based techniques into recent machine learning approaches like reinforcement learning, genetic algorithms and deep learning.

Additionally, with the big-data explosion in the current era, where data is a valuable resource and data analytics a critical factor, the importance of FCA and its generalizations in data analysis will further increase in the forthcoming years. FCA can be used for organizing data and generating meaningful insights to support important processes (see [7]).

1.3 Thesis Objectives and Scope

The prime objective of this research is to study FCA & its generalizations, and its applications in various domains with an emphasis to learning and classification problems. We propose, develop and evaluate a new classifying model called "LearnFCA", that integrates Fuzzy FCA, probability theory for learning and classifying data. The sub-objectives are as follows:

- Survey the state of art of FCA and its generalizations with emphasis to learning and classification problems.
- Discuss a new model ”LearnFCA” using Fuzzy FCA and a probabilistic approach for classifying unlabelled data into various classes.
- To propose an implementation approach and strategy for ”LearnFCA”.
- To demonstrate a model / proof of concept of the proposed architecture for ”LearnFCA”.
- To validate the proposed model by an using an experimental performance evaluation.

1.4 Research Contribution

The contributions of this thesis include:

1. *Survey of FCA and its generalizations*: Survey of various generalizations of Formal Concept Analysis and their applications.
2. *Develop LearnFCA*: Propose, Develop, implement and evaluate a new approach for learning and classification problems using FCA.
3. *Evaluate LearnFCA*: Performance Evaluation of LearnFCA

1.5 Thesis Outline

The rest of the thesis is organized in the following way:

- **Chapter 2.** We discuss the various generalizations of Formal Concept Analysis(FCA), their historical backgrounds and reproduce their theory and definitions as applicable.
- **Chapter 3.** We present a comprehensive survey on approaches and applications of FCA and its generalizations in various fields of data analysis, knowledge discovery & representation and learning & classification problems
- **Chapter 4.** We present a new novel approach "LearnFCA" for classification and learning problems using Fuzzy FCA and probabilistic methods.
- **Chapter 5.** We describe our model, architecture, experiments, methodology, data and results in greater detail.

Chapter 2

Generalizations of Formal Concept Analysis

Formal concept analysis (FCA) is a method of data analysis with growing popularity across various domains. Extensive work has been done using FCA in the past five decades in variety of domains and in varied range of applications including data analysis, information retrieval, and knowledge discovery. FCA can be understood as conceptual clustering method, which clusters simultaneously objects and their descriptions. FCA can also be used for efficiently computing association rules [8]. In other words, FCA provides an inherent integration of various components of conceptual processing and visualization of data and knowledge and it's dependencies. Since its inception, researchers have incorporated ideas from other mathematical theories into FCA to get interesting extensions which have been explored and used in a wide range of applications. With the advancement of technology, availability of different data formats and explosion of big data, the role of FCA has become more interesting; substantial research has been done and many new techniques have been developed using it's various extensions. In this section, we broadly discuss various extensions of FCA that have been developed by researchers along with their historical roots and examples.

	m_1	m_2	m_3	m_4	m_5	m_6
g_1	0.5		0.2			0.3
g_2	0.1	0.5				0.2
g_3	0.6					0.7
g_4			0.3	0.4	0.6	
g_5			0.1	0.2	0.8	
g_6				0.6		

Table 2.1: An example of fuzzy formal context with $\chi = 0.5$

2.1 Fuzzy FCA

Fuzzy set theory is an extension of classical notion of set theory where elements have degrees of membership rather than complete membership. Fuzzy set theory was introduced by Lotfi A. Zadeh [9] and Dieter Klaua [10] in 1965. Fuzzy FCA [11] uses this *fuzzy set* theory logic and generalizes FCA thereby representing object attribute membership using an uncertainty value. In a set of papers over the years [12, 13, 14, 15, 16], Belohlavek et al. have done the major derivations for fuzzy galois connections, fuzzy closure operators, fuzzy lattices and showed how fuzzy concept lattice may be used as an ordinary concept lattice. Fuzzy FCA allows representation of strengths of relationships to various degrees (from weaker to stronger) usually in a unit interval $[0, 1]$. Fuzzy FCA is useful in applications where relationships between objects and attributes are incomplete. Relationships in traditional FCA are binary, either the attribute belongs or does not belong to the object. Table 2.1 shows an example of Fuzzy FCA with objects g_1 to g_6 and attributes m_1 to m_6. In the example, the object g_3 has attributes m_1 and m_6 with a membership value of 0.6 and 0.7 respectively. Similarly, attribute m_5 belongs to objects g_4 and g_5 with membership values 0.6 and 0.8 respectively.

Definition 3 *A Fuzzy Formal Context is a five-tuple $K = (G, M, I, \mu, \chi)$,where G is a set of objects, M is a set of attributes, $I = ((GM), \mu)$ is a fuzzy set,with each pair*

$(g, m) \in I$ has a membership value $\mu(g, m)$ in $[0, 1]$ and χ the confidence threshold of the context.

Each pair $(g, m) \in I$, where I is the fuzzy set, has a membership value $\mu(g, m)$ in [0, 1]. In the example in Table 2.1 above, $\mu(g_2, m_2) = 0.5$.

For sets $G' \subseteq G$ and $M' \subseteq M$, we define two sets $G'' = \{m \in M | \forall g \in G', \mu_I(g, m) \geq \chi\}$ and $M'' = \{g \in G | \forall m \in M', \mu_I(g, m) \geq \chi\}$.

Definition 4 *A Fuzzy Formal Concept(or fuzzy concept) C of a fuzzy formal context K with a confidence threshold χ, is $C = (I'_G, M')$, where, for $G' \subseteq G$, $I'_G = (G', \mu)$, $M' \subseteq M$, $G'' = M'$ and $M'' = G'$. Each object g has a membership $\mu_{I'_G}$ defined as $\mu_{I'_G}(g) = min_{m \in M'}(\mu_I(g, m))$ where μ_I is the fuzzy function of I.*

where (g, m) is the membership value between object g and attribute m, which is defined in I. G' and M' are the extent and intent of the formal concept (I'_G, M') respectively.

2.2 Rough FCA

A Rough Set(RS) is an extension of basic set represented in terms of a pair of sets which give the lower and the upper approximation of the original set. In 1991, a Polish researcher Zdzislaw I. Pawlak first described this concept of rough set [17] as an approximation of a crisp set. Rough FCA [18, 19] incorporates rough set(RS) theory into traditional FCA to provide a generalization which finds applications to many real-world scenarios.

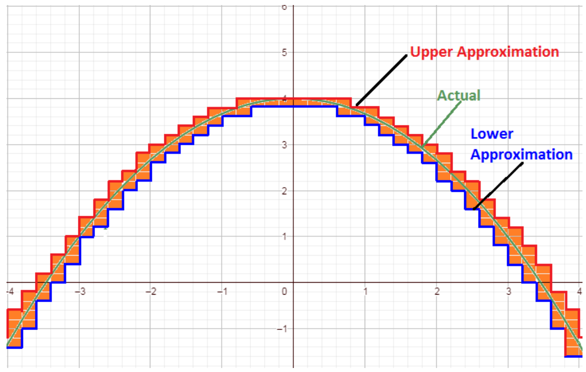

Figure 2.1: A continuous curve $y = 4 - x^2/3$ being approximated using discrete upper and lower rough set(s)

Similar to probability theory and fuzzy set theory, rough set theory is another technique for dealing with uncertain, inconsistent and imprecise information which is very evident in today's data. Figure 2.1 shows an intuition behind *rough set*. Let us consider that we want to compute the area under the curve $y = 4 - x^2/3$. Using a rough set approach for practical purposes, a discrete approximation can be obtained either using a lower-bound approach (where we sum up the area of the rectangles bounded by blue color) or a higher-bound approach (where we sum up the area of rectangles bounded by red color). Both the approaches though not be accurate, are probably easier to compute and might be practical under certain circumstances.

In rough set theory, the data for analysis consists of universe U. By modeling indiscernibility as an equivalence relation, one can partition a finite universe of objects into pair wise disjoint subsets denoted by U/P. The partition provides a granulated view of the universe. An equivalence class is considered as a whole, instead of many individuals. For an object , the equivalence class containing x is given by $[x]_P = \{y \in U|xPy\}$. Objects in $[x]_P$ are indistinguishable from x. The empty set, equivalence classes and unions of equivalence classes form a system of definable subsets under discernibility.

	m_1	m_2	m_3	m_4	m_5	m_6
g_1	Low	No	Normal	Yes	True	False
g_2	High	Yes	Very Low	Yes	True	True
g_3	Medium	No	Very High	No	False	True
g_4	High	Yes	Very Low	Yes	True	True
g_5	Medium	No	Low	Yes	False	False
g_6	Medium	No	Very High	No	False	True
g_7	Medium	No	Very High	No	False	True

Table 2.2: An example of rough set

Definition of a Rough Set Let $X \subseteq \mathbb{U}$ be a target set of a universe $\mathbb{U}$ that we wish to represent using attribute subset P; The set of objects X comprises a single class, and needs to be expressed using the equivalence classes induced by attribute subset P. In general, X cannot be expressed exactly, because the set may include and exclude objects which are indistinguishable on the basis of attributes P.

Consider the example in Table 2.2. Based on the properties m_1 to m_6, the data can be partitioned into four equivalence classes namely

$$\{\{g_1\}, \{g_2, g_4\}, \{g_3, g_6, g_7\}, \{g_5\}\}$$

Now, consider the set $X = \{g_2, g_3, g_4, g_5\}$, with the full attribute set $P = \{m_1, m_2, m_3, m_4, m_5\}$. The set X cannot be expressed exactly, because in $[x]_P$, objects $\{g_3, g_6, g_7\}$ are indiscernible. Hence, representing set X clearly such that it includes g_3 but excludes objects g_6 and g_7 becomes impossible. However, we can approximate X using only the information contained within P by constructing the P-lower and P-upper approximations of X:

$$\underline{P}X = \{x \mid [x]_P \subseteq X\}, \overline{P}X = \{x \mid [x]_P \cap X \neq \emptyset\}$$

Lower approximation (positive region). The P-lower approximation, or positive region, is the union of all equivalence classes in $[x]_P$ which are contained by (i.e., are maximal subsets of) X. In the example above, $\underline{P}X = \{g_2, g_4\} \cup \{g_5\}$. Intuitively, the lower approximation is a conservative approximation, it consists of a maximal set of objects in $\mathbb{U}/P$ that can be positively (i.e., unambiguously) classified as belonging to X.

Upper approximation (negative region). The P-upper approximation is the union of all equivalence classes in $[x]_P$ which have non-empty intersection with the target set (i.e minimal supersets of) X. In the example, $\overline{P}X = \{g_2, g_4\} \cup \{g_5\} \cup \{g_3, g_6, g_7\}$ which is the union of the three equivalence classes in $[x]_P$ that having non-empty intersection with X. The upper approximation is a liberal approximation, it consists of the minimal complete set of objects that in $\mathbb{U}/P$ that cannot be positively (i.e., unambiguously) classified as belonging to the complement ($\overline{X}$) of the target set X (but are still possibly members of X). In other words, we can say that the set $\mathbb{U} - \overline{P}X$ represents the negative region, which consists the set of objects that can be definitely not part of the target set.

Boundary region. The boundary region, given by set difference $\overline{P}X - \underline{P}X$, consists of those objects that can neither be ruled in nor ruled out as members of X.

The rough set and its accuracy. The tuple $\langle \underline{P}X, \overline{P}X \rangle$ composed of the lower and upper approximation is called a rough set; it is composed of two crisp sets, one representing a lower boundary of the target set X, and the other representing an upper boundary of the target set X. From the perspective of $\mathbb{U}/P$, the lower approximation contains objects that are members of the target set with certainty (probability $= 1$), while the upper approximation contains objects that are members of the target set with non-zero probability (probability > 0).

In [17], Pawlak provide a measure of how closely the rough set is approximating the target set. He defines the accuracy of the rough-set representation of the set X using the following:

$$\alpha_P(X) = \frac{|\underline{P}X|}{|\overline{P}X|}$$

We note that $0 \leq \alpha_P(X) \leq 1$. When the upper and lower approximations are equal (i.e., boundary region empty), then $\alpha_P(X) = 1$, and the approximation is perfect; at the other extreme, whenever the lower approximation is empty, the accuracy is zero (regardless of the size of the upper approximation).

Various definitions of rough set approximations have been proposed in literature, namely the subsystem-based, granule-based and element-based formulation [20]. They have been found to be particularly useful for rule induction and feature selection (semantics-preserving dimensionality reduction). The regions of rough sets can be interpreted as three different regions of acceptance, rejection and deferment which leads to a three-way decision making approach and many interesting applications. Rough set-based data analysis methods have been successfully applied in bio-informatics, economics and finance, medicine, multimedia, web and text mining, signal and image processing, software engineering, robotics, and engineering (e.g. power systems and control engineering) [20].

When combined with FCA, it is called as Rough Formal Concept Analysis(RFCA). $\mathbb{U}/P$ is replaced by lattice L and the sets of objects by extents of formal concepts. The extents of the resulting two concepts are the lower and upper approximations of X which are defined by:

$$\underline{l}X = \{x \mid (x, y) \in L, [x]_P \subseteq X\}, \bar{l}X = \{x \mid (x, y) \in L, [x]_P \cap X \neq \emptyset\}$$

The lower approximation of a set of objects X is the extent of formal concept $((\underline{l}X, [\underline{l}X]'))$ and the upper approximation is the extent of the formal concept

	m_1	m_2	m_3	m_4	m_5	m_6
g_1				X	X	
g_2	X	X		X	X	X
g_3						X
g_4	X	X		X	X	X
g_5				X		
g_6			X			X
g_7			X			X

	m_1	m_2	m_3	m_4	m_5	m_6
g_1			X	X	X	
g_2	X	X		X	X	X
g_3	X		X			X
g_4	X	X		X	X	X
g_5	X		X	X		
g_6	X		X			X
g_7	X		X			X

Table 2.3: Lower and upper approximation contexts using Rough Formal Concept Analysis(RFCA)

$((\bar{l}X, [\bar{l}X]'))$. $((\underline{l}X, [\bar{l}X]'))$ is the supremum of concepts where extents are subsets of X and $((\bar{l}X, [\underline{l}X]'))$ is the infimum of those concepts where extents are superset of X. Table 2.3 shows an example of upper and lower approximation contexts of data shown in Table 2.2.

A number of Rough FCA techniques [21] have been developed and very useful with large and complex data in varied range of domains namely, data mining, knowledge classification, knowledge representation and discovery, machine learning, knowledge acquisition and discovery, decision analysis, expert system, decision support system, inductive inference, conflict resolution, pattern recognition and medical diagnostics applications to handle real-world data.. In [21], the authors indicate that Rough FCA still needs to be researched thoroughly in a large number of problems, such as large data sets, efficient reduction algorithm, parallel computing and hybrid algorithms.

2.3 Monotone FCA

Deogun & Saquer [22] generalized the basic notion of formal concept to allow disjunctions in the intent and set unions in the extent. This yields in a new extension of FCA called *monotone* FCA, for which they further derive order-theoretic properties of concept hierarchies for a monotone concept.

	m_1	m_2	m_3	m_4	m_5
g_1			X	X	X
g_2	X	X	X	X	
g_3	X	X	X	X	
g_4	X				X
g_5	X				X
g_6	X				X

Table 2.4: A formal context used for monotone FCA

The authors use boolean conjunctive and disjunctive expressions on attribute subsets of a formal context and first define a notion of *feasibility* of a Boolean conjunctive expression. They also define a *monotone boolean formula* as disjunction of boolean conjuctive expressions and define its feasibility. That is, if $B_1, B_2, ..., B_n$ are Boolean conjunctive expressions, then $F = B_1 \vee B_2 \vee \cdots \vee B_n = \bigvee_{i=1}^{n} B_i$ is a monotone formula. A monotone formula $F = \bigvee_{i=1}^{n} B_i$ is said to be feasible if each B_i is feasible for $i = 1, 2, ..., n$. Otherwise, F is non-feasible. For example, $F_1 = (m_3 \wedge m_4 \wedge m_5) \vee (m_1 \wedge m_2 \wedge m_3 \wedge m_4)$ and $F_2 = m_1 \wedge m_5$ (also written as $F_1 = m_3m_4m_5 \vee m_1m_2m_3m_4$ and $F_2 = m_1m_5$) are two different feasible monotone formulas for the context given in Table 2.4.

This enables to extend the original attribute set M of a formal context (G, M, I) to a new attribute set $\mathcal{M}$ where $\mathcal{M}$ is the space of monotone formulas that can be built from M. That is, $\mathcal{M} = \{F | F = \bigvee_{i=1}^{n} B_i\}$, where each B_i is a Boolean conjunctive expression associated with $B_i' \subseteq M\}$. It is important to note that $\mathcal{M}$ contains all the monotone formulas, feasible or non-feasible, that can be generated from M. A *monotone context* and *monotone concept* is then defined as follows:

Definition 5 *A monotone context is defined as a quadruple $(G, M, \mathcal{M}, I)$ where G is a set of objects, M is a set of features, $\mathcal{M}$ is the space of all monotone formulas that can be built from M, and I is a binary relation from G to $\mathcal{M}$. For $g \in G$ and $F \in M$, gIF means that g satisfies F.*

Definition 6 *A monotone concept in the monotone context* $(G, M, \mathcal{M}, I)$ *is a pair* (A, F) *where* $A = \bigcup_{i=1}^{n}$ *is a monotone extent,* $F = \bigvee_{i=1}^{n} B_i$ *is a feasible monotone Boolean formula,* $\delta(F) = A$*,* $\gamma(A) = F$*, and each* (A_i, B_i')*, where* B_i' *is the set of features associated with* B_i*, is an elementary concept for* $1 \leq i \leq n$*.*

This generalization allows knowledge to be represented as a monotone formula and querying into an information retrieval system using disjunctions in the intent of a monotone concept. The authors also show that the set of all monotone concepts of a monotone context forms a complete lattice. Combining it with rough set theory, in [23], they discuss a novel technique to find monotone concepts whose extents are approximation of a set of objects.

2.4 Probabilistic FCA

Probability is a chance of occurrence of an event. The mathematical branch of probability theory expresses this in terms of a probability space, which is called a probability measure between 0 and 1, to a set of outcomes called the sample space. Any specified subset of these outcomes is called an event. Use of probability theory in FCA has yielded in a few different extensions of FCA in last few years, with [24, 25] as the most important ones.

In [25], the authors have integrated probability theory into traditional FCA leading to an interesting application of the membership relationship. They have extended the traditional binary membership relationship to a set of worlds by including a probabilistic measure which is a value between 0 and 1.

A probability measure $\mathcal{P}$ on a countable set W is a mapping $\mathcal{P} : \psi(W) \rightarrow [0, 1]$ such that $\mathcal{P}(\phi) = 0$, $\mathcal{P}(W) = 1$, and P is σ-additive, i.e., for all countable families $(U_n)_{n \in N}$ of pairwise disjoint sets $U_n \subseteq W$ it holds that $\mathcal{P}(\bigcup_{n \in N} U_n) = \sum_{n \in N} \mathcal{P}(U_n)$.

W_1	m_1	m_2	m_3
g_1			
g_2	X	X	
g_3			
g_4	X	X	

W_2	m_1	m_2	m_3
g_1		X	
g_2	X		
g_3		X	
g_4	X	X	X

W_3	m_1	m_2	m_3
g_1	X		
g_2	X	X	
g_3	X		
g_4		X	

Table 2.5: A probablistic formal context with $P(W_1) = 1/3, P(W_2) = 1/4, P(W_3) = 5/12$.

A world $w \in W$ is possible if $\mathcal{P}\{w\} \geq 0$, and impossible otherwise. The set of all possible worlds is denoted by W_ϵ, and the set of all impossible worlds is denoted by W_0. Obviously, $W_\epsilon \uplus W_0$ is a partition of W. Of course, such a probability measure can be completely characterized by the definition of the probabilities of the singleton subsets of W, since it holds true that $\mathcal{P}(U) = \mathcal{P}(\bigcup_{w \in U}\{w\}) = \sum_{w \in U} \mathcal{P}(w)$.

Definition 7 *A probabilistic formal context K is a tuple (G, M, W, I, P) that consists of a set G of objects, a set M of attributes, a countable set W of worlds, an incidence relation $I \subseteq G \times M \times W$, and a probability measure P on W. For a triple $(g, m, w) \in I$ we say that object g has attribute m in world w. Furthermore, the derivations in world w as operators $I(w) : \psi(G) \to \psi(M)$ and $I(w) : \psi(M) \to \psi(G)$ where*
$A^{I(w)} := \{m \in M | \forall g \in A : (g, m, w) \in I\}$ for object sets $A \subseteq G$, and
$B^{I(w)} := \{g \in G | \forall m \in B : (g, m, w) \in I\}$ for attribute sets $B \subseteq M$,
i.e., $A^{I(w)}$ is the set of all common attributes of all objects in A in the world w, and $B^{I(w)}$ is the set of all objects that have all attributes in B in w. The formal context induced by a world $w \in W$ is defined as $K(w) := (G, M, I(w))$.

A probabilistic formal context consisting of three objects g_1, g_2, g_3, three attributes m_1, m_2, m_3, and three worlds w_1, w_2, w_3 is shown in Table 2.5. We note that the object g_2 has the attribute m_1 in all three worlds, and the object g_3 has the attribute m_2 only in the world w_2. The authors provide one approach of scaling the context

$W*$	m_1	m_2	m_3
(g_1, w_1)			
(g_2, w_1)	X	X	
(g_3, w_1)			
(g_4, w_1)	X	X	
(g_1, w_2)		X	
(g_2, w_2)	X		
(g_3, w_2)		X	
(g_4, w_2)	X	X	X
(g_1, w_3)	X		
(g_2, w_3)	X	X	
(g_3, w_3)	X		
$(g_4, w_3$		X	

Table 2.6: The scaled probabilistic formal context of Table 2.5

by integrating the object-world to generate a traditional formal context using the definition below.

Definition 8 *(Scaling). Let K be a probabilistic formal context. The certain scaling of K is the formal context $K^{\times} := (G \times W, M, I^{\times})$ where $((g, w), m) \in I^{\times}$ iff $(g, m, w) \in I$, and the almost certain scaling of K is the subcontext $K_{\epsilon} := (G \times W_{\epsilon}, M, I_{\epsilon}^{\times})$ of $K^{\times}$.*

A scaled version of probabilistic context of Table 2.5 is shown in Table 2.6

Other approaches include [24], where Demin et al. have introduced another approach for probabilistic extensions and appropriate probabilistic variants of formal concepts and implications. An inductive probabilistic approach to formal concept analysis (FCA) is proposed in [26] where probabilistic concepts with predictive force is defined. The authors also discuss ability to handle nonclassified objects, eliminating random attributes and generating concepts robust to noise. In [27], Deogun et al. present a logic model for Knowledge discovery in databases based on an integrated approach of Bacchus probability logic and formal concept analysis which they use to deduce previously unknown and potentially useful patterns in databases.

Obs	m_1	m_2	m_3	m_4	m_5	m_6
t_1				X	X	
t_2	X	X		X	X	X
t_3						X
...	...	...	..	..	..	..
t_5				X		

	m_1	m_2	m_3
(g_{1t_1})			
(g_{2t_2})	X	X	
(g_{3t_3})			
(g_{4t_4})	X	X	
(g_{5t_5})		X	

Table 2.7: Temporal FCA contexts: (i) CTSOTs where points of time (observations) are used as objects in a formal context, (ii) TCSS where entities at a particular time granule are used as objects

	m_1	m_2	m_3	m_4	m_5	m_6
(obs_{1t_1})	0.1	0.61	0.91		0.73	
(obs_{2t_2})			0.5	0.23		0.11
(obs_{3t_3})	1.0		0.2		0.34	
(obs_{4t_4})	0.1	0.63			0.71	0.21

Table 2.8: Temporal Fuzzy FCA Context with threshold $\chi = 0.5$

2.5 Temporal FCA

Temporal Concept Analysis (TCA) is the theory of temporal phenomena integrated into Formal Concept Analysis (FCA). TCA was introduced and developed by the Wolff [28], the main idea was to be able to represent a state of an object at a certain time in a temporal system into FCA. States are defined as formal concepts and 'points of time' are generalized to 'time granules', interpreted as 'pieces' of time needed for the realization of measurements. The authors in [28] discuss three important developments of TCA, namely Conceptual Time Systems with actual Objects and a Time relation (CTSOTs), Temporal Conceptual Semantic System (TCSS) and Temporal Conceptual Semantic Systems (TCSSs) among others. CTSOTs use temporal objects and a time relationship, TCSS uses the notion of a distributed object which may occupy at each time granule a certain volume and TRSS integrates recent developments in theory of Temporal Conceptual Semantic Systems with conceptual scaling of formal context. Table 2.7 shows two example contexts for CTSOT and TCSS.

Temporal Fuzzy FCA. A very interesting and recent development in the field of TCA was the notion of Temporal Fuzzy FCA(TFCA) by De Maio et al. [7], where the authors proposed integration of Temporal phenomena with Fuzzy Concept Analysis. They applied TFCA on a distributed real-time computation system for big data stream analysis in a smart city context to organize the knowledge of various aspects of the city and generate temporal patterns of it's evolution. The proposed approach is similar to one in TCSSs above and index the time-stamped objects by adding a time variable to the object of the fuzzy formal context, namely g_{t_i} with $g \in G$. g_{ti} represents the object $g \in G$ at time t_i, where t is the time variable and t_i precedes t_j if $i \leq j$. All the time intervals $t_i \in 1, ..., n$, and the number of observations n made on a particular object over time form a partial order relation.

Let M be the set of formal attributes, and m_j be the j^{th} attribute, where i and j are integers.

Definition 9 *The intension of an object O at time t_i is the set of all attributes of that particular object at time t_i.*

$$i\{O_{ti}\} = \{m_j \in M\}$$

Definition 10 *An evolution of an object, $Ev(O)$, is an ordered set of sets containing all the intensions that this particular object from an arbitrary initial time t_0 to a certain time t_i. The intension of the concept at time t_{i+1} is added to the resulting evolution set when we consider the evolution up to time t_{i+1}:*

$$t_0 \leq t_i, i\{O_{ti}\} \in Ev_{t_0 - t_i}(O)$$

Definition 11 *A Timed Fuzzy Lattice L^t is a pair (L, E^t) where $L = (C, \leq)$ is a*

Fuzzy Lattice and E^t is a set of temporal edges which are 2-element subsets of C.

Given $L^t = (L, E^t)$, the time relationship is reflected in temporal edge $e^t_{ij} \in E^t$ between two concepts $(C_i, C_j) \in C$ which exists iff there exist two objects $g_{t_s} \in C_i$ and $g_{t_k} \in C_j$ such that the time t_s precedes the time t_k.

Definition 12 *A temporal path is defined as a path $\pi = (C_s, \cdots, C_t) \in C \times C \cdots \times C$, such that exists a temporal edge $e^t_{ij} \in E^t$ for $s \leq i \leq j \leq t$.*

In the Temporal Fzuzzy FCA context, the observation parameters are used as attributes and observation times are used as objects. Table 2.1 shows one such example. By using an incremental algorithm for generating timed fuzzy lattice and taking snapshots at various points in time, the authors show that a temporal path can be constructed to track the growth of various city parameters. This can be further used in assisting and supporting smart city decision making processes.

2.6 Logical Concept Analysis

Logical Concept Analysis(LCA) is a generalization of Formal Concept Analysis proposed by Ferre in [29]. The main idea is that the attribute set in traditional FCA are now logical expressions of arbitrarily any kind. The authors derive all order theoretic properties of this extension of FCA and show that the lattice generated is isomorphic to the original lattice.

The authors reformulate traditional FCA by replacing the traditional context (O, A, I) by $(O, 2^A, i)$, where 2^A is the power-set of A and i is a mapping from O to 2^A defined by $i(o) := \{a \in A | (o, a) \in I\}$. Also, $(o, a) \in I \Leftrightarrow i(o) \supseteq \{a\}$. Then, 2^A can be considered as a logic where $\supseteq$ is the deduction relation, $\cap$ is the disjunctive operation, and $\cup$ is the conjunctive operation. The power set of set of attributes 2^A is

object	**description**
g_1	a
g_2	b
g_3	$a \vee c$
g_4	$c \wedge (a \vee b)$

	$m_1 = a$	$m_2 = b$	$m_3 = c$	$m_3 = a \vee c$	...
g_1	X				...
g_2		X			...
g_3	X			X	...
g_4	X	X			...

Table 2.9: An Logical Concept Analysis(LCA) table and its transformed context to generate the logical concept lattice

replaced by a set of arbitrary logical formulas $\mathcal{L}$, to which are associated a deduction relation $\models$, a disjunctive operation $\dot{\vee}$, and a conjunctive operation $\dot{\wedge}$.

Definition 13 *(Logical context) A (formal) context is a triple $(\mathcal{O}, \mathcal{L}, i)$ where:*

1. *$\mathcal{O}$ is a finite set of objects,*
2. *$\langle \mathcal{L}; \models \rangle$ is a lattice of formulas, whose supremum is $\dot{\vee}$, and whose infimum is $\dot{\wedge}$; $\mathcal{L}$ denotes a logic whose deduction relation is $\models$, and whose disjunctive and conjunctive operations are respectively $\dot{\vee}$ and $\dot{\wedge}$,*
3. *i is a mapping from $\mathcal{O}$ to $\mathcal{L}$ that associates to each object a formula that describes the intention of the object.*

Definition 14 *Let $(\mathcal{O}, \mathcal{L}, i)$ be a context, $O \subseteq \mathcal{O}$, and $f \in \mathcal{L}$. Two applications σ and τ are defined as follows:*

$$\sigma : 2^{\mathcal{O}} \to \mathcal{L}, \sigma(O) := \dot{\bigvee}_{o \in O} i(o)$$

$$\tau : \mathcal{L} \to 2^{\mathcal{O}}, \tau(f) := \{o \in \mathcal{O} | i(o) \models f\}$$

σ and τ form a Galois connection. In other words, the Galois connection is between sets of objects (extent) and logical formulas (intent).

Definition 15 *(Logical concept) In a context* $(\mathcal{O}, \mathcal{L}, i)$ *, a concept is a pair* $c = (O, f)$ *where* $O \subseteq \mathcal{O}$ *, and* $f \in \mathcal{L}$ *, such that* $\sigma(O) = f$ *and* $\tau(f) = O$*.*

The set of all concepts that can be built in a context $(\mathcal{O}, \mathcal{L}, i)$ is denoted by $C(\mathcal{O}, \mathcal{L}, i)$, and is partially ordered by $\leq^c$ to get a ordered set $\langle C(\mathcal{O}, \mathcal{L}, i); \leq^c \rangle$ which is a complete lattice. Table 2.9 shows an example of LCA table and its reformulated context which can be used to generate the LCA lattice.

Other approaches include logical scaling of object-attribute-value relationships [30], epistemic extension based on modal logic AIK [31] and generating classification rules based on lattice [32].

2.7 Relational Concept Analysis

Relational Concept Analysis(RCA) was first introduced by Huchard et al. [33] where the authors emphasized the role and applications of classical FCA to complex relational data. They propose a new extension that takes as a collection of contexts and inter-context relations as inputs and yields a set of lattices whose concepts are linked by relations. The main idea behind Relational concept analysis (RCA) is to allow processing of multi-relational datasets, i.e., with multiple sorts of individuals with its own set of attributes, and relationships between them.

In RCA [33, 34], data are organized within a structure composed of a set of contexts $K = \{\mathcal{K}_\rangle\}$ and of a set of binary relations $R = \{r_k\}$, where $r_k \subseteq O_i \times O_j$, O_i and O_j being sets of objects (respectively in $\mathcal{K}_i$ and $\mathcal{K}_j$). The structure (K, R) is called a relational context family (RCF) and can be compared to a relational database schema, including both classes of individuals and classes of relations.

Definition 16 *A relational context family* R *is a pair* (K, R)*, where* K *is a set of* *contexts* $\mathcal{K}_i = (O_i, A_i, I_i)$, *R is a set of relations* $r_k \subseteq O_i \times O_j$ *where* O_i *and* O_j *are the object sets of the formal contexts* $\mathcal{K}_i$ *and* $\mathcal{K}_j$ *.*

A relation $r \subseteq O_i \times O_j$ can be seen as a set-valued function $r : O_i \to 2^{O_j}$. Two functions are defined on relation sets in RCF, domain and range:

- $O = \{O_i/O_j \in \mathcal{K}_i = (O_i, A_i, I_i), \mathcal{K}_i \in K\}$

- $r_k : O_i \to 2^{O_j}$

- $dom : R \to O$ with $dom(r_k) = O_i$

- $ran : R \to O$ with $ran(r) = O_j$,

- $rel : K \to 2^R andrel(\mathcal{K}_i) = \{r_k | dom(r_k) = O_i\}$

where O is the set of all object sets in the RCF, $O = \{O | \mathcal{K} = (O, A, I) \in K\}$. Moreover, an auxiliary function maps a context into the set of all relations whose domain corresponds to the object set of the context:
$rel : K \to 2^R; rel(\mathcal{K} == (O, A, I)) = \{r | dom(r) = O\}$.

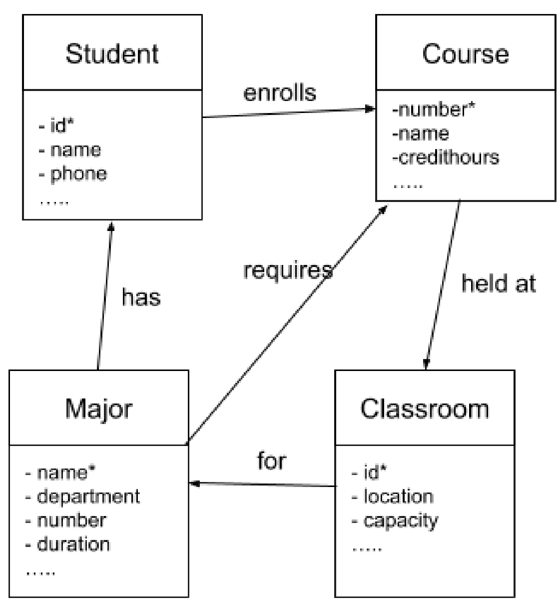

Figure 2.2: An example relational schema for Relational Concept Analysis(RCA)

Lets take an example student enrollment schema in Figure 2.2. Students "enroll" in Courses which are "held at" different classrooms. A major "requires" a set of courses and "has" a set of enrolled students. Each entity has its own set of attributes or properties. Using RCA approach, this schema can be scaled (or transformed) into a family of contexts, some of them are object/attribute contexts and others object/object contexts which are as follows:

1. $K_{Student1} \in$ Student $\times$ name, $K_{Student2} \in$ Student $\times$ phone
2. $K_{Course1} \in$ Course $\times$ name, $K_{Course2} \in$ Course $\times$ credithours
3. $K_{Major1} \in$ Major $\times$ department, $K_{Major2} \in$ Major $\times$ duration
4. $K_{Classroom1} \in$ Classroom $\times$ location, $K_{Classroom2} \in$ Classroom $\times$ capacity
5. $K_{enrolls} \in$ Student $\times$ Course
6. $K_{held} \in$ Course $\times$ Classroom
7. $K_{for} \in$ Classroom $\times$ Major
8. $K_{has} \in$ Major $\times$ Student
9. $K_{requires} \in$ Major $\times$ Course

Table 2.10 and 2.11 show few examples of object-attribute and object-object contexts generated from the relational schema. A scale attribute combining a relation r with a formal concept $c = (X, Y)$ from the lattice L_j is assigned to an object $o \in O_i$ whenever $r(o)$ is "correlated" with the extent of c. In other words, given a relation r such that $dom(r) = O_i$ and $ran(r) = O_j$, the target set of concepts will correspond to a concept lattice of the context underlying O_j.

	n_1	n_2	n_3	n_4
s_1				X
s_2		X		X
s_3			X	
s_4	...	...	..	..
s_5		X		

	p_1	p_2	p_3	p_4
s_1				X
s_2	X	X		X
s_3				
s_4	...	...	..	..
s_5				X

Table 2.10: RFCA object-attribute contexts for $K_{Student1}$ and $K_{Student2}$

	co_1	co_2	co_3	co_4
s_1		X		X
s_2		X		X
s_3	X		X	
s_4	...	...	..	..
s_5	X	X		

	cl_1	cl_2	cl_3	cl_4
co_1			X	X
co_2	X			
co_3			X	
co_4	...	...	..	..
co_5		X		

Table 2.11: RFCA object-object contexts for $K_{enrolls}$ and K_{held}

Applying FCA to relational context families(RCF) results in a family of lattices due to expansion of contexts through relational scaling and a iterative process that is repeated till the lattices evolve completely.This allows for a way of representing the concepts and relations extracted with RCA in the framework of a description logic.

The authors in [33] also state and define separate modes of relational scaling called narrow (for $\forall r.C$) and wide (for $\exists r.C$). They implement RCA in a tool called *Galicia* [35] which offers efficient tools for knowledge and software engineering allowing reasoning and problem-solving. To address the scalability issue for RCA, recently Bazin et al. [36] proposed an approach to generate only a concept and its neighbour concepts at each navigation step during the generation of the extended concept lattices.

2.8 Other Generalizations

2.8.1 Triadic FCA

The triadic approach to concept analysis (TrCA) was introduced by Wille and Liehman in [37]. TrCA is an extension of Formal Concept Analysis which adds an additional dimension to Formal concept analysis; resulting in a triadic relation connecting objects, attributes, and conditions.

Definition 17 *(triadic context) A triadic context is a quadruple $\langle X, Y, Z, I \rangle$ where X, Y , and Z are nonempty sets, and I is a ternary relation between X, Y, and Z, i.e. $I \subseteq X \times Y \times Z$. X, Y , and Z are interpreted as the sets of objects, attributes, and conditions, respectively and I is interpreted as the incidence relation. That is, $\langle x, y, z \rangle \in I$ is interpreted as: object x has attribute y under condition z and x, y, z are related by I.*

On similar principles, a triadic concept is defined and the structure of the triadic concepts is analysed. Wille et al. also show that the concepts can be organized as *"complete trilattices"* which is isomorphic to traditional FCA lattice. Recent developments in TrFCA include [38] where the authors provide a general approach for developing an unifying framework for concept-forming operators, [39] where fuzzy attributes are handled, [40] where analysis is done on data with graded attributes and [41] where a tri-clustering approach is used.

2.8.2 Pattern Structures and FCA

Pattern structures with FCA generalize traditional FCA definitions to include graphs and numeric intervals. Although pattern structures have been used since 1990's, it was formalized by Kuznetsov in [42]. Kuznetsov studies the analysis of complex data with pattern structures, indicating that pattern structures can be more efficiently computed and visualized compared to traditional FCA based methods.

Let G be a set of objects, let $(D, \sqcap)$ be a meet-semilattice (of object descriptions) and let $\delta : G \rightarrow D$ be a mapping. Then $(G, \underline{D}, \delta)$ with $\underline{D} = (D, \sqcap)$ is called a *pattern structure*, and if the set

$$\delta(G) := \{\delta(g) | g \in G\}$$

generates a complete subsemilattice $(D_\delta, \sqcap)$, of $(D, \sqcap)$. Thus each $X \subseteq \delta(G)$ has an infimum $(\sqcap X)$ in $(D, \sqcap)$ and $(D_\delta, \sqcap)$ is the set of these infima. Elements of D are called patterns and are ordered by subsumption relation $\sqsubseteq$: given $c, d \in D$ one has $c \sqsubseteq d \Leftrightarrow c \sqsubseteq d = c$.

A pattern structure $(G, \underline{D}, d)$ gives rise to two operators,

$A^\Box = \sqcap_{g \in A} \delta(g)$ for $A \subseteq G, g \in A$ and

$d^\Box = \{g \in G | d \sqsubseteq \delta(g)\}$ for $d \sqsubseteq D$.

These operators form a Galois connection between the powerset of G and $(D, \sqsubseteq)$. Pattern concepts of $G, \underline{D}, d$ are pairs of the form (A, d), $A \sqsubseteq G, d \in D$ such that $A^\Box = d$ and $A = d^\Box$ with extent A and intent d. The *pattern lattice* can then be generated by starting from descriptions in an arbitrary order $(P, \leq)$ taking as $(D, \sqcap)$, the lattice of all order ideals of $(P, \leq)$.

Pattern structures have been used on sets of graphs and vectors of intervals [43], version spaces in FCA [44] to express version spaces in FCA. Some other applications have been to mine gene expression data [45] and decision tree induction using interval pattern structures [46].

2.8.3 Possibility Theory View of FCA

Possibility theory is a mathematical theory for dealing with certain types of uncertainty and is an alternative to probability theory. Professor Lotfi Zadeh [9, 47] first introduced possibility theory in 1978 as an extension of his theory of fuzzy sets and fuzzy logic. Didier Dubois and Henri Prade further contributed to its major development [48]. Possibility theory enables to retrieve an enlarged perspective for the traditional formal concept analysis.

Possibility theory is characterized by four set functions - a possibility measure Π, a dual measure of necessity $\mathcal{N}$, a measure of "actual or guaranteed possibility" $\cdot$ and a dual measure of "potential necessity or certainity" ∇. A possibility distribution is defined on a universe U which represents a value of some quantity similar to a membership function of a fuzzy set E in U.

A possibility-theoretic view of formal concept analysis [48] uses particular set-valued counterparts of the four main set-functions of possibility theory. The Galois connection is based on the actual (or guaranteed) possibility function, where each object in a concept has all properties of its intent, and each property is possessed by all objects of its extent. The concepts then form a lattice or a set of sub-lattices analogous to traditional FCA.

Possibility theory with FCA has been studied in [48] for decomposing and characterizing contexts into minimal sub-contexts, [49] for uncertainity handling based on possibilistic representation framework, [50] for handling fuzzy data.

Chapter 3

Applications of Formal Concept Analysis and its generalizations

With ever-growing and never-ending complexity of techniques and approaches, it becomes important to classify all work-done in a field using a variety of models and methods to successfully describe the current state of research. This chapter surveys the state-of-art of approaches and applications of Formal Concept Analysis and its extensions from existing literature. We use a taxonomy to classify all relevant work into three major categories which are defined as follows:

1. **Data Analysis**: This category includes data clustering (Table 3.1), data mining (Table 3.2), data extraction & estimation (Table 3.3 & 3.4), data analysis and recommendation (Table 3.5), data optimization (Table 3.6), data visualization & personalization (Table 3.7) and data quality management (Table 3.8).

2. **Knowledge Management**: Concerning with overall transformation of knowledge from one form to other, in this category, we include discovery of knowledge, knowledge representation, learning and storage into ontologies (Table 3.9 & 3.10).

3. **Learning and Classification**: These approaches which are also applicable

to streaming or big data, include techniques to learning from existing complete data and generating efficient models which can then be used to make predictions on un-classified data. (Table 3.11 & 3.12).

3.1 Approaches in Data Analysis

3.1.1 Clustering

Authors	**Methodology**
Freeman(1996) [51]	Use the Galois structure of containment among cliques to produce clusters (communities) that are intuitive, and consistent with ethnographic descriptions.
Maille (2005) [52]	Use FCA to cluster incident reports based on their outcome and initial situation.
Myat and Hla (2005) [53]	Cluster textual web documents based using FCA on the terms they contain.
Okubo and Haraguchi (2006) [54]	Cluster web documents using FCA and provide a conceptual meaning for each cluster.
Kuznetsov et al. (2007) [55]	Find scientific communities using FCA and using a stability measure to select pertinent concepts.
Roth, Obiedkov, and Kourie (2006) [56], Roth, Obiedkov, and Kourie(2008) [57]	Use FCA with nested line diagrams and concept stability to identify communities of researchers in scientific papers on a well-defined topic.

Table 3.1: FCA Applications in Data Clustering

3.1.2 Data Mining

Authors	**Methodology**
Eisenbarth et al. (2003) [58]	Construct mapping of features to source-code using a semi-automatic technique and FCA.
Tonella and Ceccato (2004) [59]	Aspect identification is done using FCA by means of dynamic code analysis of execution traces
Busch and Richards (2004) [60]	Use FCA for the modelling and analysis of psychological questionnaire data.
Maille, Statler, and Chaudron (2005) [52]	Use FCA in the developing a Kontex system to identify causes in aeronautical incident reports.
Formica (2006,2008) [61, 62]	Analyze similarity between ontology concepts using FCA
Girault (2008) [63]	Use concept lattices to devise an unsupervised method to analyze the relations between named entities and their syntactic dependencies observed in a training corpus.
Lounkine, Auer, and Bajorath (2008) [64]	Use FCA to identify molecular fragment combinations by analyzing structure-activity relationships between compounds and biological activities.
Rouane-Hacene, Toussaint, and Valtchev (2009) [65] and Villerd, Toussaint and Lillo-Le Louët (2010)[66]	Analyze case reports to mine adverse drug reactions in pharmacovigilance data using formal concept analysis.
Ebner et al. (2010) [67]	Analyzed twitter content generated using FCA during and after a scientific conference.
Poelmans et al. (2010) [68]	Analyze patient care activity data by combining FCA with Hidden Markov Models.
Stumpfe,Lounkine,and Bajorath (2011)[69]	Used FCA for computational selectivity studies and analysis of structures of chemical compounds and their selectivity towards certain targets.
Egho, Jay, Raissi, and Napoli (2011) [70]	Apply sequential pattern mining to sequences of hospitals (represented by attributes) and patients (represented by objects) hospitalized during their cancer treatment.
Endres, Adam, Giese, and Noppeney (2012) [71]	Analyzed fMRI scans using FCA of a human subject while he was viewing 72 gray-scale pictures of animate and inanimate objects during a target detection task.
Elzinga, Wolff, Poelmans, Viaene, & Dedene (2012) [72]	Used Temporal Concept Analysis(TCA) and Temporal Relational Semantic Systems(TRSS) to the analysis of chat conversations.

Table 3.2: FCA Applications in Data Mining

3.1.3 Extraction and Estimation

Authors	Methodology
Besson et al. (2004, 2005)[73]	Introduced a D-miner algorithm to compute concepts under constraints. (e.g. minimal and maximal frequency)
Mens and Tourwé (2005) [74]	Used FCA to analyze source code of a system for relevant concepts.
Sklenar, Zacpal, and Sigmund (2005) [75]	Used FCA to find dependencies between demographic data and degree of physical activity in epidemiological questionnaire data .
Old (2006) [76]	Disambiguating homographs using FCA with illustrative examples, using data from Roget's Thesaurus
Breu, Zimmermann, and Lindig (2006)[77]	Extracted aspects from Eclipse by analyzing where developers added code to the program over time.
Fan and Xiao (2007) [78]	Proposed an FCA-based method for ontology mapping.
Beydoun, Kultchitsky, and Manasseh (2007) [79]	Developed a system KAPUST to capture user trails as they search the internet and learn search trails.
Del Grosso, Penta, and Guzman (2007) [80]	Identify pieces of functionality to be exported as services from database-oriented applications. Dynamically extract database queries during the execution of the application.
Belohlavek, Sklenar, Zacpal, and Sigmund (2007, 2011) [81, 82]	Build further on the work of Sklenar et al. (2005) and Sigmund, by aggregating respondents and using fuzzy values to indicate the relative frequency of attributes in the aggregated objects.
Sato, Okubo, Haraguchi, and Kunifuji (2007) [83]	Use FCA to analyze similarities in the time series data related to medical test results.
Zhao, Halang, and Wang (2007) [84]	Present an ontology mapping method based on rough FCA and a proposed rough similarity measure.

Table 3.3: FCA Applications in Data Extraction and Estimation - I

Authors	**Methodology**
Kim, Hwang, and Kim (2007b) [85]	Used FCA for extracting internet blogs, extracted tags become attributes and bloggers become the objects of the context.
Lu and Zhang (2008) [86]	Conflict detection and elimination between two equivalent concepts in different source ontologies with different definitions of value and cardinality restriction.
Cellier et al. (2008) [87]	FCA is used in combination with association rules for fault localization in software source code.
Molloy et al. (2008) [88]	Use FCA for efficiently extracting roles from user-permission and user-attribute information.
Motameny et al. (2008) [89] Gebert et al. (2008) [90]	Use an FCA-based model to identify combinatorial biomarkers of breast cancer.
Fenza et al. (2008, 2009)[91, 92]	Present a method using fuzzy FCA for supporting the user in the discovery of semantic web services.
Bertaux et al. (2009) [93]	Describe a method to identify ecological traits of species based on the analysis of their biological characteristics.
Dufour-Lussier et al. (2010)[94]	Identify ingredients which can be used to replace an ingredient in another recipe using FCA.
Poelmans et al. (2010, 2011, 2012) [95, 96, 97], Elzinga et al. (2010) [98]	Use FCA to identify human trafficking and terrorism suspects from observational police reports.
Falk et al. (2010, 2011) [99, 100]	Use FCA and translation to extract most relevant verb-frame associations in language data.
Messai et al. (2011) [101]	Use FCA to identify critical patient-related characteristics related to physician non-compliance with Clinical Practice Guidelines (CPG).
Keller et al. (2012) [102]	Use FCA lattice for gene associations to evaluate the complexity of the relationships among diseases, and to identify concepts for further functional analysis.
Priss et al. (2012) [103]	Use FCA to gain insight into the conceptual structure of difficulties in learning processes of students.

Table 3.4: FCA Applications in Data Extraction and Estimation -II

3.1.4 Analysis and Recommendation

Authors	Methodology
Cho et al. (2004) [104]	Developed a web search system that reuses keywords and web pages previously entered and visited by other persons and provide recommendations.
Rudolph (2004) [105]	Propose an incremental method based on FCA which uses empirical data to systematically generate hypothetical axioms about the domain of interest in ontologies.
Zhou et al. (2004) [106]	Use an FCA based model to mine association rules from web usage logs and use a recommendation engine which matches them with the user's recent browsing history.
Chi et al. (2005)	Construct ontological knowledge bases for digital archive systems.
Hauff and Deogun (2007) [107]	Concept lattices are used for disjoint clustering of transactional databases and several heuristics are developed to tune the support parameters used in the algorithm.
Colton and Wagner (2007) [108]	FCA is used in combination with mathematical discovery tools to better facilitate mathematical discovery.
Ignatov and Kuznetsov (2008)[109]	Used FCA to develop a recommender system for suggesting potentially interesting advertisement terms that can be bought.
Quan et al. (2006) [110]	The authors apply a FCA based method to build an ontology which can be used in a web-based help-desk application.
Jay et al. (2008) [111]	Apply ice-berging and stability based concept lattices to choose better medical treatment trajectories for cancer patients.
Huang (2008) [112]	Use Rough Concept analysis for marine ontology to generate recommendations.
Wang et al. (2009) [113]	Use FCA to compute the Concept- Concept similarity, the Concept-Ontology similarity between two agent crawlers.
Dau and Knechtel (2009) [114]	Apply FCA in combination with Description Logics to capture the RBAC constraints and for deriving additional constraints.
Solesvik and Encheva (2009) [115]	Use FCA as a quantitative instrument for partner selection in the context of collaborative ship design.
Ignatov et al. (2011) [116]	Built lattice based taxonomies to represent the structure of student assessment data to identify the most stable student groups.
Romashkin et al. (2011) [117]	Analyzed university applications using lattice-based taxonomies derived from entrants' decisions about undergraduate programs.

Table 3.5: FCA Applications in Data Analysis & Recommendation

3.1.5 Optimization

Authors	Methodology
Cole et al.(2000,2001, 2003) [118, 119, 120], Eklund (2004) [121]	Use concept lattices for email search, discovery and visualization, a Conceptual Email Manager was developed.
Stumme et al. (2001) [122], Cure and Jeansoulin (2008) [123]	Propose a solution for merging of ontologies using FCA.
Carpineto et al. (2004) [124]	System optimization for information retrieval using ice-berging of lattices.
De Souza et al. (2004a, 2004b) [125, 126]	Use ontology merging for aligning, evaluating concept similarities to process user queries.
Richards (2004) [127]	Describes a method based on Ripple-Down rules and FCA in the biology domain on four knowledge-bases about a plant.
Cole et al. (2005) [128]	FCA is used to conceptually analyze relational structures in software source code and to detect unnecessary dependencies between software parts.
Koester (2005) [129], Koester (2006) [130]	FooCA: Improving web search engine results with contexts and concept hierarchies.
Dau et al. (2008) [131] Ducrou and Eklund (2007) [132]	Build Search-sleuth using a formal context which contains the result of the query as objects and the terms found in the title & summary of each result as attributes.
Yang et al.(2008b) [133]	Build a topic-specific web crawler with concept similarity context graph which gathers only particular pages.
Du et al. (2009) [134]	Association rule mining optimization for web pages using FCA.
Boutari (2010) [135]	Use FCA to expand short texts with additional terms to reduce context sparseness.

Table 3.6: FCA Applications in Data Optimization

3.1.6 Visualization and Personalization

Authors	Methodology
Priss (2004) [136], Priss and Old(2010) [137]	Discuss how FCA can be used to visually represent lexical databases, i.e. organized collections of words in electronic form.
Kim and Compton (2006) [138]	Combined lattice based browsing with conceptual scales to extend the search functionality by reducing the complexity of the visualization.
Wermelinger, Yu, and Strohmaier (2009)[139]	Use FCA lattices to visualize the relations between the software artifacts and to indicate developers for bug-fixing.

Table 3.7: FCA Applications in Data Visualization and Personalization

3.1.7 Quality Management

Authors	**Methodology**
Choi, et al. (2006) [140] and Choi et al. (2008) [141]	Created concept lattices from micro array data of genes and compared to each other using a distance metric.
Choi et al. (2008) [141] and Priss and Old (2010) [137]	Investigate how FCA can be used for Roget's thesaurus and for visualizing WordNet.
Pan et al.(2009) [142]	Use FCA for comparison of radiology report content before and after adoption of PACS system.
Le Grand et al. (2009) [143]	Use FCA for complex systems analysis and compare different topic maps with each other both in terms of content and structure.
Jiang, Pathak, and Chute (2009) [144]	Audit the completeness and correctness of the International Classification of Disease (ICD) codes using FCA.
Sertkaya (2009) [145]	Describe OntoComp, which supports ontology engineers in checking an OWL ontology for relevant information about the application domain and in extending the ontology if required.
Jiang and Chute (2009) [146]	Used FCA to analyze the completeness and correctness of SNOMED, a controlled vocabulary for the medical domain.
Kaytoue, et al. (2009 [45]), Kaytoue, et al. (2011) [147]	FCA in combination with interordinal scaling is compared to pattern structures based on interval vectors for mining gene expression data and extracting co-expressed genes.
Kirchberg et al. (2012) [148]	Performed an in-depth comparison of all performance aspects of analyzing large amounts of sematic web data, obtained from the Internet, in real- time.

Table 3.8: FCA Applications in Data Quality Management

3.2 Approaches in Knowledge Discovery, Representation and Learning

3.2.1 Building Knowledge Base and Ontologies

Authors	**Methodology**
Jiang et al. (2003)[149]	Use FCA in combination with NLP for semi-automatically building a Japanese ontology in the cardiovascular medicine domain.
Soon and Kuhn (2004) [150]	Use FCA for producing task-oriented ontologies.
Cimiano et al. (2004) [151]	Discuss and present several examples on how FCA can be used to support ontology engineering and how ontologies can be exploited in FCA applications.
Chi et al.(2005) [152]	construct ontological knowledge bases for digital archive systems
Hwang et al. (2005) [153]	Use FCA for the construction of ontologies in the domain of software engineering.
Xu et al. (2006) [154]	FCA is used to build an event ontology from a set of textual documents.
Richards (2006) [155]	Build personal and ad hoc ontologies using FCA which may help gaining understanding of the research domain.
Quan et al. (2006) [156]	Apply FCA to build an ontology which can be used in a web-based help-desk application.
Fang et al. (2007) [78]	Integrate FCA with Protege to build a knowledge sharing platform, containing information about the acupuncture points from traditional Chinese medicine.
Wollbold et al. (2008) [157]	Derive a knowledge base using FCA using mRNA and protein concentrations consisting of a set of transition rules between states.
Huang (2008) [112]	Proposed rough FCA for semi-automatically constructing a marine domain ontology.
Wollbold et al. (2009) [158]	Use FCA and attribute exploration for building a knowledge base about a gene regulatory network of a bacterium.
Xu et al. (2009) [159]	Contemplate on how FCA can be used to build a computer network management information specification ontology.

Table 3.9: FCA Applications in Building Knowledge base and Ontologies

3.2.2 Enhancing/Optimizing Knowledge Base and Ontologies

Authors	**Methodology**
Richards and Malik (2003) [160]	Use a method to restructure knowledge bases containing classification rules in the domain of chemical pathology.
Cimiano, Hotho, and Staab (2005)[161]	Applied FCA and NLP to textual data from the tourism and finance domain to automatically learn ontological concept hierarchies.
Gamallo, Lopes, and Agustini (2007) [162]	Use technical articles in computational linguistics and a list of 175 terms to extract lexico-syntactic contexts using FCA and find which terms co-occur with which lexico-syntactic contexts.
Kim et al. (2007) [138]	Propose a method to extract ontological elements from OWL source code and create a context family of five kinds of contexts.
Kiu and Lee (2008) [163]	Use FCA for managing existing ontological knowledge instead of building an ontology.
Bendaoud, Toussaint, and Napoli (2008)[164]	Propose an FCA-based system for semi-automatically enriching an initial ontology from a collection of texts in the astronomy domain.

Table 3.10: FCA Applications in Enhancing Knowledge base and Ontologies

3.3 Approaches in Learning and Classification Problems

Authors	**Methodology**
Jean-Gabriel Ganascia(1987)[165]	CHARADE: constructs classification rules based on k-DNF expressions built from the generated lattice.
Carpineto et al. (1993) [166]	GALIOS: classifies by computing similarity between instances and coherent maximal concepts.
Liquiere et al. (1993) [167]	LEGAL: searches for coherent maximal concepts and uses majority vote for classification
Oosthuizen, G (1994) [168]	GRAND: a classification strategy to use the most specific rules containing largest number of examples and the smallest number of attributes.
Sahami, M. (1995) [169]	RuLearner: employs an algorithm to generate rules based on having to find recursively the minimum set of attributes to classify maximum set of instances.
Njiwoua et al. (1999) [170]	CIBLE: adopts heuristic techniques to generate only the pertinent and useful concepts without constructing the entire lattice.
Xie et al. (2002) [171]	CLNN & CLNB: integrated Naive Bayes and Nearest Neighbor into concept lattice nodes to classify new instances.
Ganter and Kuznetsov (2003) [172]	Combined version spaces from machine learning with FCA lattices to generate new classifiers.
Ganter and Kuznetsov(2000) [173], Kuznetsov (2004, 2004) [174, 175]	Present a classification model of learning from positive and negative examples using FCA lattice and and graph-theoretic interpretations of concept-based classification rules.
Ganter et al. (2004) [176] Kuznetsov et al. (2005) [177]	Introduced pattern structures and combined with JSM method in machine learning to develop their classification model.
Auon-Allah et al. (2006) [178]	Use FCA lattice to validate rules generated from a meta-classifier for mining very large distributed databases.
Ricordeau (2003) [179] Ricordeau and Liquiere (2007) [180]	Introduced Q-concept learning algorithms, by generalizing reinforcement algorithms in machine learning using the standard FCA lattice.
Rudolph (2007) [181]	Present an encoding strategy for knowledge processing using FCA and develop a classification method using a neural network.

Table 3.11: FCA Applications in Learning & Classification Problems - I

Authors	**Methodology**
Wen et al. (2007) [182]	Propose a classification algorithm by using an ontology learning method and clustering on fuzzy concept lattice
Chang et al. (2008) [183]	Propose a document classifier system by using FCA on existing knowledge ontology and integrating with Naive Bayes Classifier.
Girault (2008) [63]	Use FCA concept lattice mining to develop an unsupervised classifier for named entity annotation.
Tsopze et al. (2009) [184]	CLANN: a two-class supervised classifier, which uses the FCA concept lattice to directly build the network architecture of a neural network.
Carpineto et al. (2009) [185]	Build a classifier using FCA lattice and integrating with Support Vector Machines (SVM).
Belohlavek et al. (2009) [186] Lindig, (2000) [187]	Proposed a classifier by constructing decision trees using nodes of concept lattice generated from input data.
Outrata (2010) [1]	Use FCA and Boolean Factor Analysis to build a classifier using decision tree induction.
Visani et al. (2011) [188]	NAVIGALA: uses a navigation based strategy on FCA lattice to develop a supervised classification algorithm for noisy data.
Kuznetsov (2013) [189]	Use FCA pattern structures for knowledge discovery and classification in Big data.
Meddouri et al. (2014) [190]	Propose new parallel classifiers based on sub-lattice generated from relevant concepts and Dagging of Nominal Classifier.
Kashnitsky et al. (2015) [191]	RMCS: Introduce a Multi-classifier system based on recommendations using neighborhood nodes in FCA lattice.
Ali (2018) [192]	Use FCA and Bagging for learning and generation of classification rules.
Mezni et al. (2018) [193]	Propose a collaborative filtering based recommendation system using Fuzzy FCA.
Fray et al. (2019) [194]	Implemented a Distributed Classifier Nominal Concepts(CNC) (based on FCA) for classification.

Table 3.12: FCA Applications in Learning & Classification Problems - II

Chapter 4

LearnFCA - A novel approach for learning and classification

4.1 Introduction

Classification problems traditionally fall into two main categories, *supervised* and *unsupervised* learning. Machine learning is a scientific study of constructing computer programs that can learn from data and improve with experience. FCA has originally been used to exploit the relationship between a set of objects and their properties, thereby learning predictable patterns from available data to build a knowledge base of the domain. Use of FCA combined with various machine learning models and domain expertise has yielded in some interesting results. These machine learning models include mining classification rules [165, 166, 167, 168, 178], decision tree induction [186, 1], artificial neural networks [170, 181], nearest neighbor algorithms [171, 188], naive bayes classifier[171, 184, 188], support vector machines [185], version spaces [172] , JSM method [43],q-concept learning [179, 180], dagging [190] and bagging [192].

In this chapter, we first discuss some important techniques that are used for transforming real data before it can be processed by FCA. We also introduce an intuitive and naive technique of using FCA lattice for classification. We then discuss various approaches from existing literature. Finally, we propose a new model "LearnFCA" which integrates class information into the FCA lattice and uses probabilistic techniques to classify instances. The architecture and implementation of "LearnFCA" is presented with an emphasis on our classification strategy.

4.2 Handling Real Data

Real data is complex and real objects seldom have binary attributes. Hence, we need to pre-process real data before it can be analyzed by FCA. In this section, we discuss some very well-known approaches used for pre-processing real data.

4.2.1 Conceptual Scaling

Probably the most popular approach, conceptual scaling [6] is a process of transforming *multi-valued* data to *one-valued* formal context by extending the set of attributes. Techniques include nominal scaling (non-numeric values) , ordinal scaling (ranking numeric values), inter-ordinal scaling (partitioning values into intervals) and bi-ordinal scaling (partitioning values with ranking). An example of nominal scaling and ordinal scaling is presented in Figure 4.1 and Figure 4.2 respectively.

Name	*body temp.*	*gives birth*	*fourlegged*	*hibernates*
cat	warm	yes	yes	no
bat	warm	yes	no	yes
salamander	cold	no	yes	yes
eagle	warm	no	no	no
guppy	cold	yes	no	no

Name	*bt cold*	*bt warm*	*gb no*	*gb yes*	*fl no*	*fl yes*	*hb no*	*hb yes*
cat	0	1	0	1	0	1	1	0
bat	0	1	0	1	1	0	0	1
salamander	1	0	1	0	0	1	0	1
eagle	0	1	1	0	1	0	1	0
guppy	1	0	0	1	1	0	1	0

Figure 4.1: Nominal scaling of multi-valued context taken from [1]

	age	$marks$
s_1	12	20
s_2	15	35
s_3	17	70
s_4	13	50

	$a < 15$	$m > 45$
s_1	X	
s_2		
s_3		X
s_4	X	X

Figure 4.2: Ordinal scaling of a numeric multi-valued context

4.2.2 Context Reduction

Reducing the size of a formal context can result in faster lattice generation and improved lattice structure. One such technique is to remove multiple objects with exact same attributes since it doesn't affect the lattice structure. In the formal context in Table 1.1, objects "Electrical Engg" and "Computer Engg" can be combined together into one object without affecting the lattice structure. Another common technique is to use nested-line diagrams to partition the context and lattice into sub-context(s) and sub-lattice(s). Larger lattices can be decomposed into smaller sub-lattices which can then be easier to interpret and analyze. Figure 4.3 shows an example of decomposition of a lattice generated from a planet context from [2].

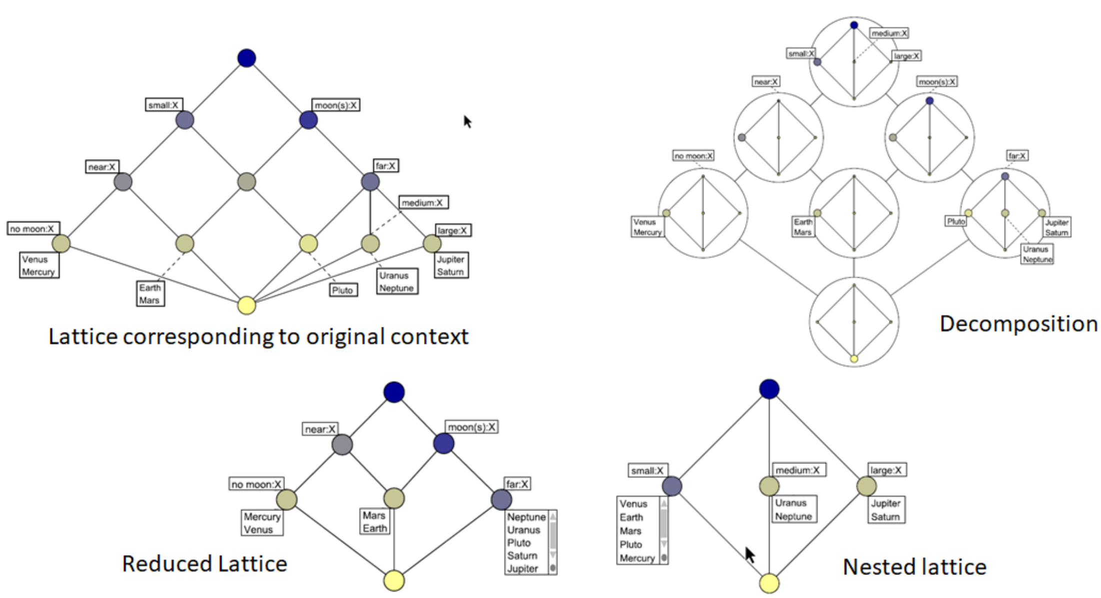

Figure 4.3: Decomposing lattices into sub-lattices (nested line diagram) in a planet context example from [2]

4.2.3 Noise Removal

This technique involves reducing the lattice structure by removing some of the formal concepts that donot meet a specified criteria. One popular method is ice-berging [195] where concepts with low extent size are dropped. Another method is to define a *concept stability* measure based on how strongly a concept depends on it's objects (or attributes) and ignore concepts below a certain threshold. [196]

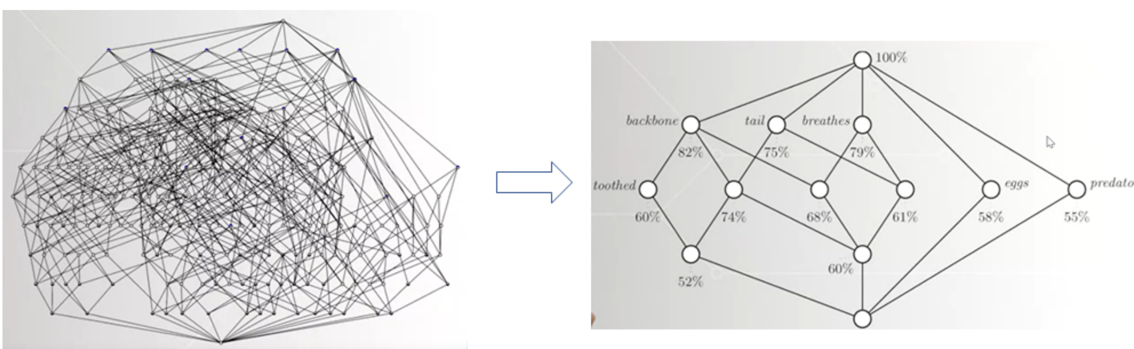

Figure 4.4: Ice-berg lattice generated from modified UCI machine learning zoo dataset by removing concepts that cover less than half of the object set [3]

4.3 Using FCA Lattice to classify - Intuitive Example

Name	*body temp.*	*gives birth*	*fourlegged*	*hibernates*	*mammal*
cat	warm	yes	yes	no	yes
bat	warm	yes	no	yes	yes
salamander	cold	no	yes	yes	no
eagle	warm	no	no	no	no
guppy	cold	yes	no	no	no

Name	*bt cold*	*bt warm*	*gb no*	*gb yes*	*fl no*	*fl yes*	*hb no*	*hb yes*	*mammal*
cat	0	1	0	1	0	1	1	0	yes
bat	0	1	0	1	1	0	0	1	yes
salamander	1	0	1	0	0	1	0	1	no
eagle	0	1	1	0	1	0	1	0	no
guppy	1	0	0	1	1	0	1	0	no

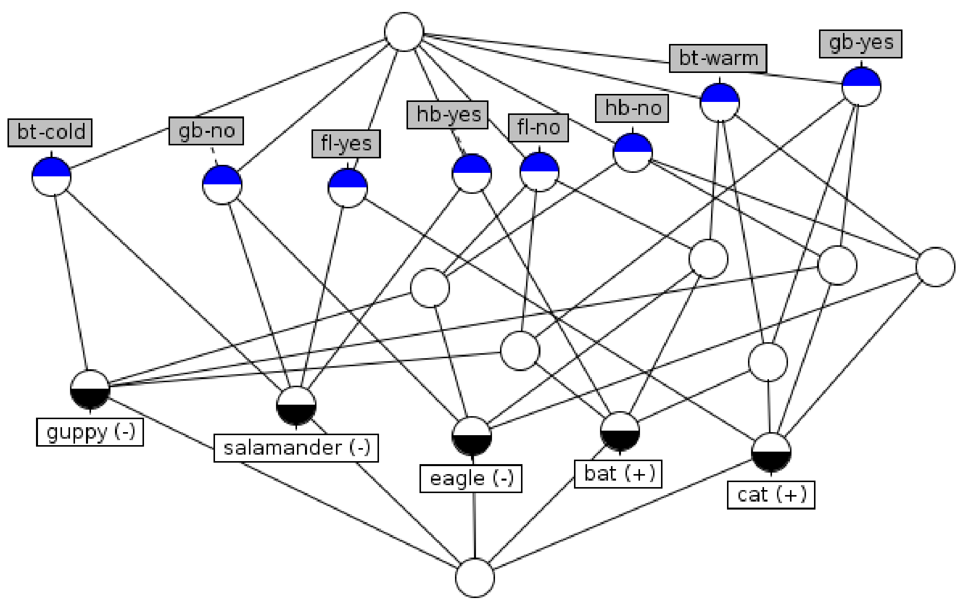

Figure 4.5: An example formal context and its concept lattice taken from [1]

Figure 4.5 shows a small formal context defined by a set of animals representing *objects* and a set of properties exhibited by them representing attributes of the domain. At first, the *attributes* are multi-valued and are scaled to binary values. The generated lattice consists of 21 concepts. The top and the bottom nodes are trivial concepts and other concepts are identified by the labels of objects and attributes on them or inherited based on the edges connecting the nodes.

Algorithm 1: A naive algorithm to generate classification rules from a concept lattice

---

Input: A formal context $K = (G, M, I, S)$

Output: Classification rules of form $A \rightarrow S', S' \in \{yes,no\}$

Generate a list of concepts C and the concept-lattice L.

while *all objects are not classified:* **do**

  From the top, move right and then down to find a concept c that classifies most number of objects of a class but not the other classes.

  Include c in C'.

  Remove the classified objects and their concept from C.

Generate classification rules based on C'.

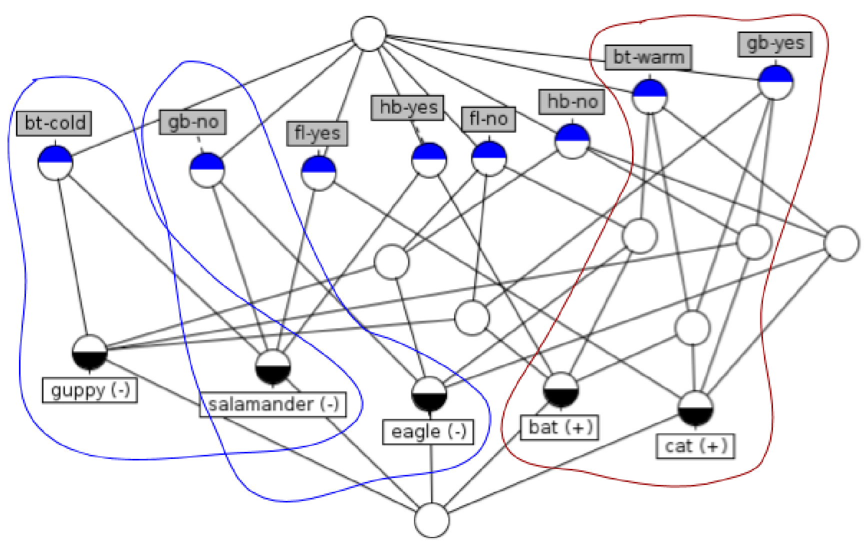

Figure 4.6: An application of the naive algorithm (Algorithm 1) on lattice to generate classification rules

A naive algorithm to generate classification rules from the concept lattice is described in Algorithm 1. The five instances (two positive and three negative) are classified into two categories (mammal or not-mammal). When Algorithm 1 is applied on the lattice (Figure 4.6), the generated classification rules are:

1. $\{bt\text{-}cold\} \rightarrow no$
2. $\{gb\text{-}no\} \rightarrow no$
3. $\{bt\text{-}warm, gb\text{-}yes\} \rightarrow yes$

4.4 Related Work

FCA has been used with machine learning as early as 1980's with work mostly concentrated around generating classification rules from the concept lattice and using them to classify new instances. Probably, the earliest model of use of FCA for classification can be seen in CHARADE [165] developed in 1987. It constructs classification rules based on k-DNF expressions built from the generated lattice and claims to have better performance than the basic machine learning ID3 algorithm. Carpineto et al. [166] developed GALIOS in 1993, which computes the similarity between a new instance and a set of coherent/maximal concepts, uses a majority vote strategy to classify the instance to the most similar class. LEGAL [167] proposed in 1993, uses a similar inductive method, searches for coherent/maximal concepts and uses majority vote in the set of pertinent regularity to classify new examples. GRAND, developed in 1994 [168] employs a classification strategy to use the most specific rules which contain the largest number of examples and the smallest number of attributes. The authors select and choose the three best rules to classify a new instance. RuLearner(1995) [169] formally defines and employs an algorithm to generate rules based on having to find recursively the minimum set of attributes to classify maximum set of instances of a specific category. CIBLE(1999) [170] adopts heuristic techniques to generate only the pertinent and useful concepts without constructing the entire lattice. It combines induction and k-NN, and one of the three distance measures: Mahalanobis, Manhattan and Euclidean to classify an new instance.

In 2002, Xie et al. [171] developed new composite classifiers (CLNN and CLNB) by integrating Naive Bayes and Nearest Neighbor into the concept lattice nodes, applying constraints and a new voting strategy to classify a new instance. Similar to [170], they built only partial lattices using heuristic techniques for better efficiency of their classification algorithm. Ganter & Kuznetsov [172] in 2003, combined the model of version spaces in machine learning with FCA to generate classifiers that are closed under conjunction and apply it to an example from predicitive toxicology. The authors mention in certain cases, the classifiers can be too restrictive, and suggest to use sets of hypotheses in terms of patterns structures as a possible remedy. In another series of papers by Ganter & Kuznetsov [174, 175, 173], the authors present a classification model of learning from positive and negative examples using FCA lattice and graph-theoretic interpretations of concept-based classification rules (hypothesis). Later in 2004 [176], they extended their work by introducing pattern structures and combine with JSM method in machine learning and test their classification model on Predictive Toxicology datasets.

Aoun-Allah and Mineau (2006) [178] use FCA lattice to validate rules generated from a meta-classifier for mining very large distributed databases. Ricordeau & Liquiere [179, 180] introduce Q-concept learning algorithms, by generalizing reinforcement algorithms in machine learning using the standard FCA lattice. In [181], Rudolph uses FCA to present an encoding strategy for knowledge processing data and provide a method to support the neural-symbolic learning cycle in a 3-feed forward neural network. Wen, Liu and Yan(2007) [182] propose a classification algorithm by using an ontology learning method, clustering on fuzzy concept lattice and apply it on UCI Machine learning datasets. In [183], the authors propose a document classifier system by using FCA on existing knowledge ontology and integrating with Naive Bayes Classifier with an average effectiveness of 89%. Girault(2008) [63] use FCA concept lattice mining to present an unsupervised classifier for named entity annotation to improve the existing supervised classification process. Tsopze et al. [184] propose CLANN, a two-class supervised classifier, which uses the FCA concept lattice to directly build the network architecture of a neural network. The authors believe that this approach is helpful when prior knowlege of data is not available and show the soundness and efficiency using various experiments on UCI datasets.

In 2009, Caprineto et al. [185] use FCA for pre-processing and generating a document lattice and integrate it with Support Vector Machines(SVM) to build a text based classifier. They show that the proposed method is better than the standard SVM when very little training data are available. Later in the same year, Belohlavek et al. [186] proposed a classifier by constructing decision trees using nodes of concept lattice generated from input data and experimentally evaluate with standard benchmark datasets. Outrata [1] in 2010, used formal concept analysis in input data preprocessing to generate formal concepts using boolean factor analysis and build a classifier using decision tree induction, They evaluate their performance with standard decision tree induction algorithms ID3 and C4.5. In 2011, Visani, Bertet, Ogier [188] proposed NAVIGALA, which uses a navigation based strategy on FCA lattice to develop a supervised classification algorithm for noisy data. The algorithm selects relevant concepts from the huge amount of data, which then is processed by a k-nearest neighbors or the Bayesian classifier. Some quantitative and qualitative evaluations are also presented.

Kuznetsov (2013) [189] use pattern structures as an extension of FCA for knowledge discovery and classification in big data. They state that classification using pattern structures combined with other approaches can result in a polynomial complexity algorithm which is useful for big data. In 2014, Meddouri et al. [190] propose new parallel classifiers based on on sub-lattice generated from relevant concepts and Dagging of Nominal Classifier. They evaluate their performance on UCI Machine learning datasets and also compare their results with existing classifiers. Kashnitsky et al. (2015) [191] introduce a multi-classifier system based on recommendations using neighborhood nodes in FCA lattice. They discuss their approach with an example and present results on UCI machine learning datasets. Ali (2018) [192] used FCA and Bagging for learning and generation of classification rules. In 2018, Mezni et al. [193] propose a collaborative filtering based recommendation system using Fuzzy FCA and evaluate their approach. Fray et al. [194] implemented a distributed Classifier Nominal Concepts(CNC) based on FCA for classification in 2019. They evaluate the efficiency of their approach by comparing the results to sequential single-node version.

A chronological list of the related work is presented in Table 3.11 & 3.12.

4.5 Overview and Definitions

4.5.1 FCA with Classes

Below, we present a formal definition of traditional FCA with integrated class information:

Definition 18 *(context) A Classed Formal Context is a four-tuple* $K = (G, M, I, S)$*, where* G *is a set of objects,* M *is a set of attributes,* I *is a binary function from* G *to*

M and S is a set of classes. Each pair $(g, m) \in I$ has a membership value in $\{0, 1\}$, with each object $g \in S$ belonging to a particular class $s \in S$.

In Figure 4.5, $G = \{$*cat, bat, salamander, eagle, guppy* $\}$, $M = \{$*bt cold, bt warm, gb no, gb yes, fl no, fl yes, hb no, hb yes* $\}$, $S = \{$*yes, no*$\}$.

Definition 19 *(concept) A Formal Concept with Class C_c of a formal context K is $C_c = (G', M', S')$ where for $G' \subseteq G$, $M' \subseteq M$, $G'' = M'$ and $M'' = G'$. Each concept C_c has a associated class $s \in S$ which is determined by some classification strategy ζ.*

Two formal concepts with class (Figure 4.6) are $C_{c_1} = (\{$*guppy*$\}$,$\{$*bt cold, gb yes, fl no, hb no*$\}$, *no*) and $C_{c_2} = (\{$*cat*$\}$,$\{$*bt warm, gb yes, fl yes, hb no*$\}$, *yes*). Intuitively, this means that the lattice node with the object set $G = \{$*guppy*$\}$ has been assigned a *no* class using certain classification strategy ζ. One example for ζ could be to use the conjunction of classification rules generated earlier. Classification strategies could vary from being simple to complex.

For formal context lattices with large number of nodes, classification rules and strategies can get fairly complex and it becomes difficult to assign a concept node to a particular class s. One way to resolve this is to assign a *probability vector* $\mathcal{P}$ to the concept where each item p_i is the fractional value corresponding to the class s_i, $s_i \in S$.

Definition 20 *(concept probability vector) A concept probability vector $\mathcal{P}$ for a concept C is a probability vector $\{p_1, p_2, p_3 ... p_n\}$, with each element corresponding to a class $s \in S$, $p_i \in [0, 1]$, $n = |S|$ such at $\sum p_i = 1$.*

4.5.2 Fuzzy FCA with Classes

In this section, we extend the notion of FCA with Classes to Fuzzy FCA with Classes by modifying the membership function to be from a fuzzy set $[0, 1]$. To address the difficulty of assigning classes to *concepts* in the fuzzy lattice, we use a concept probability vector $\mathcal{P}$ as discussed in the previous section.

Definition 21 *A Fuzzy Formal Context with Class (FFC_c) is a six-tuple $K = (G, M, I, S, \mu, \chi)$, where G is a set of objects, M is a set of attributes, $I = ((GM), \mu)$ is a fuzzy set, with each pair $(g, m) \in I$ has a membership value $\mu(g, m)$ in $[0, 1]$, S is a set of classes with each object $g \in G$ belonging to a particular class s, and χ the confidence threshold of the context.*

Definition 22 *A Fuzzy Formal Concept with Class (or classified concept) C_c of a fuzzy formal context K with class and a confidence threshold χ, is $C_c = (I'_G, M', \mathcal{P})$, where, for $G' \subseteq G$, $I'_G = (G', \mu)$, $M' \subseteq M$, $G'' = M'$ and $M'' = G'$. Each object g has a membership $\mu_{I'_G}$ defined as $\mu_{I'_G}(g) = min_{m \in M'}(\mu_I(g, m))$ where μ_I is the fuzzy function of I. $\mathcal{P}$ is a probability vector of the concept C representing the chances of being classified into various classes in S.*

Figure 4.7 shows a fuzzy concept lattice with probability vectors associated with each concept node.

	m_1	m_2	m_3	m_4	s
g_1		0.6	0.4	0.1	s_1
g_2	0.4	0.3		0.7	s_3
g_3	0.1		0.4	0.8	s_3
g_4	0.6				s_2
g_5		0.7	1.0	0.6	s_1
g_6	0.7		0.3	0.5	s_3

Table 4.1: A fuzzy formal context with class information

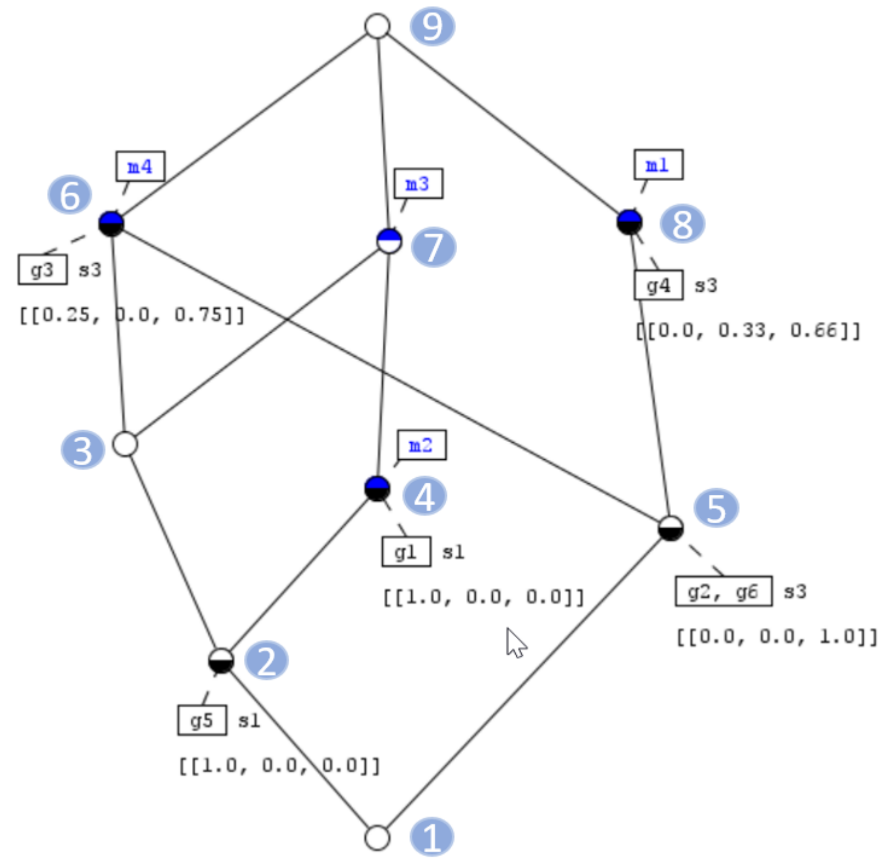

Figure 4.7: Fuzzy Concept Lattice of Context in Table 4.1 with probability concept vectors

4.5.3 Classification Strategy ζ

First, we calculate the probability concept vectors by defining $p_i \in \mathcal{P}$ of a concept C_c as the ratio of no of objects in it's extent that belong to s_i and it's extent size.

$$p_i = \frac{\text{No of objects in Extent}(C_c) \text{ in class } s_i}{\text{Extent Size}(C_c)}$$

The concept lattice in 4.7 shows probability vectors associated with each concept. For example, for node 6 with extent $O = \{g_2, g_3, g_5, g_6\}$, intent $A = \{m_4\}$; $p_1 =$ No of objects in Extent in class s_1 / Extent Size = 1/4 = 0.25. Similarly, $p_2 = 0.0$ and $p_3 = 0.75$;

Next, we present Algorithm 2 to classify all individual objects in the given context. The main idea is to traverse through the fuzzy lattice for each object (instance) o and find the relevant concept C_o that contains the object and has the maximum extent size. The predicted class of o is then assigned as the class that has the maximum value in the probability vector $\mathcal{P}_c$ for C_o.

Algorithm 2: ClassifyUsingLattice($FuzzyL$)

Input: A Fuzzy Lattice $fuzzyL$ generated using Fuzzy FCA
Output: All instances and their predicted class

Initialize $insMap \leftarrow new$ empty;
foreach *concept* fc *in* $fuzzyL$ **do**
    foreach *object* o *in* $fc.getExtent()$ **do**
        $n \leftarrow fc.getExtentSize()$;
        $m \leftarrow insMap.get(o).getSize()$;
        if $m \leq n$ **then**
            $insMap.set(o, n, fc.getPV())$;
foreach o, n, pv *in* $insMap$ **do**
    Output "Predicted class of o is :" max(pv);

Algorithm 2 initializes an empty instance map "$insMap$" which is gradually filled with the instances o, their maximum extent size n, and the resultant probability

Object	*Relevant Concept*	*Probability Vector*	*Predicted Class*	*Actual Class*
g_1	4	$<1.0,0.0,0.0>$	s_1	s_1
g_2	5	$<0.0,0.0,1.0>$	s_3	s_3
g_3	6	$<0.25,0.0,0.75>$	s_3	s_3
g_4	8	$<0.0,0.33,0.66>$	s_3	s_2
g_5	2	$<1.0,0.0,0.0>$	s_1	s_1
g_6	5	$<0.0,0.0,1.0>$	s_3	s_3

Table 4.2: Results after applying strategy ζ to fuzzy lattice in Figure 4.7

vector (*pv*) for each instance. The results of applying Algorithm 2 to the fuzzy lattice in Figure 4.7 is shown in Table 4.2. The algorithm classifies 5 out of 6 objects successfully (an accuracy of 83.33%).

4.6 LearnFCA - Architecture and Implementation

LearnFCA Model(LFM) has been developed from the java based Concept Explorer tool (*conexp*) [197] and the *conexpng* tool [198]. Both of tools have the core FCA functionality of processing contexts, generating lattices and implication rules. These tools have been modified and adapted to support Fuzzy FCA, and to construct a fuzzy lattice that can be used for classification. LFM has the capability to process classes related information and classify instances based on a probabilistic strategy described in Algorithm 2. The key characteristics of LFM are:

1. Java based implementation based on *conexp* and *conexpng*.

2. Ability to support Fuzzy FCA and process classes related information.

3. A text based classifier that works using a probabilistic strategy, reports accuracy for large datasets and has the ability to randomize and select training/test set for experimentation.

A copy of the implementation can be obtained from https://git.unl.edu/ssamal/conexpng or https://github.com/sksamal/conexpng

4.6.1 Fuzzy Lattice Construction

Algorithm 3: GenFuzzyLattice ($fuzzyC$)

Input: A given Fuzzy Context $fuzzyC$ (G, M, I, χ)
Output: A Fuzzy Lattice $fuzzyL$ with fuzzy concepts, class information and class probabilities

```
Initialize clsMap ← new empty ;
Initialize trnMap ← new empty;
fuzzyC ←CreateFuzzyContext(fuzzyC) ;
foreach object o in G do
    UpdateClassMap(clsMap, o.class) ;
    if isTraining(o) then
        UpdateTrainingMap(trnMap, o) ;
fuzzyL ←CreateLattice(fuzzyC,th) ;
Initialize nc ←size(clsMap);
foreach fuzzyConcept fc in fuzzyL do
    Initialize pv ← [0.0]*nc;
    foreach class c in clsMap do
        pv[c] ← clsMap.getCount(c) / fc.getExtentSize();
    fc.setPV(pv) ;
return fuzzyL
```

First, we use a few new data structures to extend the capability of *conexpng* tool to support fuzzy contexts and storing classes. *th* is used for the threshold of the fuzzy context. *clsMap* and *trnMap* are used for storing classes and whether a given instance is a training or a test instance. A vector *pv* is used for each lattice concept in the fuzzy lattice for storing the classification probability values.

The steps for construction of fuzzy lattice are detailed in Algorithm 3. We initialize the threshold *th* to a default value of 0.5. The data structure $clsMap$ stores the mapping of object to their classes as observed in the given context. $trnMap$ is a data structure to optionally separate out the training objects. The lattice is generated using standard FCA process and stored in $fuzzyL$. The next step involves traversing through each concept in the fuzzy lattice and adding a probability vector for each concept based on classification strategy discussed in Section 4.5.3.

4.6.2 Object Classification Process

Classifying instances (objects) to various classes is done using the fuzzy lattice in a similar way as done in 4.5.3. The classification process is detailed in Algorithm 2. An instance map $insMap$ of all objects and empty probability vectors is first initialized. We traverse through the concepts in the fuzzy lattice, and for each object in the concept, $insMap$ is updated with the resultant probability vector if required. Finally, we loop through each object in $insMap$ and output the predicted class associated with the maximum value in the probability vector.

Chapter 5

LearnFCA - Experimental Evaluation

This chapter describes in detail how we apply LearnFCA to various data, the setup and methodology we used to perform the various experiments. We also discuss results, their interpretation and future work that can be done in this direction.

5.1 Introduction and the big picture

The use of intelligent computing to assist clinical decision making and medical procedures can be seen since early 1960's, however its use has still been profoundly limited to date due to various reasons. Healthcare procedures and clinical work-flows are known to be inherently complex. Clinical decision making often involves processing an enormous range of relevant data. Multiple terminologies and nomenclature exist in medical domain that adds to the complexity. Recent years has seen the emergence of big data, increasing risk of errors in diagnostic procedures all with a need of high accuracy in medical diagnosis procedures. Hence, the use of advanced computing technologies to assist clinicians has become almost indispensable. On the other hand, recent applications of artifical intelligence have been proven very successful in complex domains like self-driving cars, natural language processing and biological systems engineering.

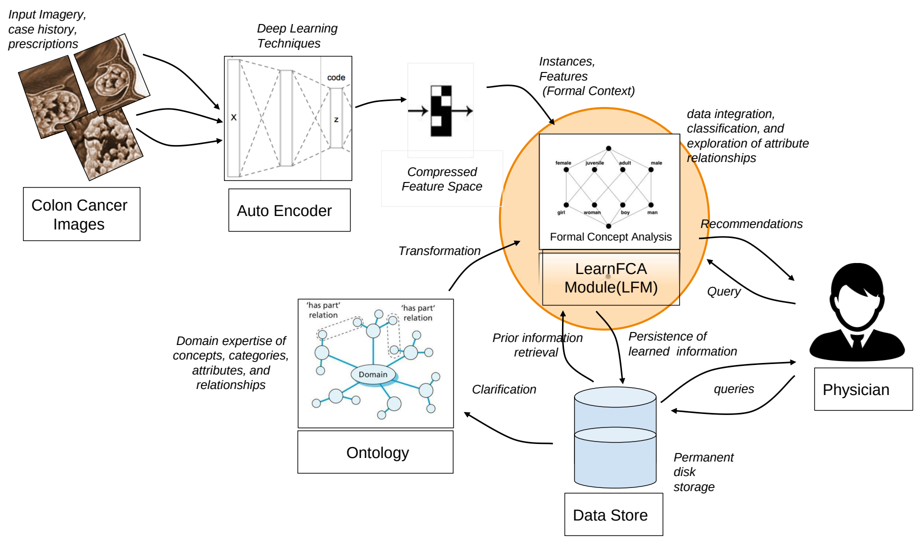

Figure 5.1: LearnFCA Model in a proposed CDSS system

"LearnFCA" is proposed to be an integral part of a clinical decision support system(CDSS) framework for physicians as shown in Figure 5.1. The entire project aims to design and implement an intelligent computing system that would assist physicians (doctors) with decision making (treatment plan, diagnosis, other recommendations) and provide recommendations for clinical evaluation and treatment. The project uses deep learning techniques to extract and incrementally learn vital data characteristics from medical images and use LearnFCA Model (LFM) to classify, and finally provide recommendations to physicians to assist in diagnostic procedures and clinical work-flows.

5.2 Experimental Setup

5.3 Setup

We evaluate "LearnFCA" on encodings (matrices) generated from images obtained from various datasets. For the FCA context, each encoding is an object and each feature is an attribute. Originally, each encoding has 64 attritbutes. Experiments were run on two types of encodings, *binary* (values are either 0 or 1) and *floating* (values are between 0 to 1). Attributes were scaled from 64 to 512 using a simple binning strategy explained in Algorithm 4. Encodings are obtained from three datasets, MNIST [199], OMNIGLOT [200] , and GDC portal cancer images [201] .

Dataset	Instances	Classes
MNIST[**199**]	60000	10
OMNIGLOT[**200**]	14000	1623
ColonCancer [**201**](**GDC Portal**)	1041	2

Figure 5.2: Datasets used for LearnFCA experiments

5.3.1 Methodology

The entire process is composed of three steps: extending the features (optional), generation of fuzzy lattice and classification using lattice all of which is depicted in Algorithm 5. The first step involves extending the original features (attributes) of size 64 to a higher dimension (128-256) using Algorithm 4 (*ExtendFeatures*) if necessary. This is done by reducing the range of each attribute. A single attribute value between 0 to 1 can be extended to four attributes by dividing the ranges to 0 to 0.25, 0.25 to 0.5, 0.5 to 0.75 and 0.75 to 1.0 as shown in Table 5.1 and Table 5.2.

The next step involves using the extended encodings as input for a formal context and generating a fuzzy lattice using Algorithm 3 (*GenFuzzyLattice*) . The generation process involves tracking of training & test instances, generating fuzzy concepts and updating the probability vectors within each lattice concept. The last and final step involves traversing the lattice, using the classification strategy ζ and predicting the best class for each instance. This is detailed in Algorithm 2 (*ClassifyUsingLattice*).

Algorithm 4: ExtendFeatures (E, n)

Input: A matrix of encodings $E[i \times f]$ with f features
Output: A matrix of encodings $E'[i \times f']$ with f' features $(|f'| = |f| * n)$

`/* Initialize encodings of size` $f * n$ `*/`
Initialize $E' \leftarrow$ empty ;
$n' \leftarrow |f| * n$;
foreach *encoding* e *in* E **do**
  Initialize $e' \leftarrow [0.0] * n'$;
  `/* Assign the feature to appropriate bin in e' */`
  foreach *feature* $e[j]$ *in* e **do**
    $bid \leftarrow int(e[j] * n)$;
    $e'[(j-1) * n + bid] \leftarrow e[j]$;
  Add *e'* to E';
return E'

---

	m_1	m_2
g_1	0.7	0.1
g_2	0.3	0.2
g_3	0.6	0.5
g_4	0.8	0.9

Table 5.1: Sample input data for Algorithm 4

	m_{11}	m_{12}	m_{13}	m_{14}	m_{21}	m_{22}	m_{23}	m_{24}
g_1			X		X			
g_2		X			X			
g_3			X			X		
g_4				X				X
	[0-0.25]	(0.25-0.5]	(0.5-0.75]	(0.75-1.0]	[0-0.25]	(0.25-0.5]	(0.5-0.75]	(0.75-1.0]

Table 5.2: Inter-ordinal scaling example to extend features in sample data (Table 5.1)

Algorithm 5: Algorithm to classify encodings based on Fuzzy lattice

Input: A set of encodings matrix $Enc[i \times f]$, with i instances and f features. $i = i_1$(training instances) $+i_2$(test instances) and $F = f_1$ (features) $+f_2$ (classes)

Output: instances and their classified classes

1 $EncX \leftarrow$ExtendFeatures(Enc, n);

2 $FuzzyL \leftarrow$GenFuzzyLattice$(EncX)$;

3 ClassifyUsingLattice$(FuzzyL)$;

5.3.2 Results

MNIST Image Encodings [199]

MNIST is a database of handwritten digits, having a training set of $60,000$ examples, and a test set of $10,000$ examples and 10 classes. It is widely used for training and testing in the field of machine learning and classification problems. The digits have been size-normalized and centered in a fixed-size image of $28x28$ pixels each. We apply our classification algorithm to 100 to 500 of these images. With original set of encodings (64 attributes), our algorithm classifies about 25% of the instances accurately. On scaling the attributes to 512, the accuracy goes to as high as 96.4% (Table 4.2).

Omniglot Data Encodings [200]

The Omniglot data set contains 1623 characters (classes) spanning 50 different alphabets and about 14000 instances. However, it has very few examples per class. We apply our classification algorithm by selecting about 1000 to 5000 images from the dataset. The accuracy is low(1.5%- 2.5%) with original set of encodings with 64 attributes, but when scaled to 512 attributes, the accuracy increases to be in the range $39\% - 55\%$ (see Table 4.2). One of the limitations of our model is the inability to handle larger number of instances.

Colorectal Cancer Images Dataset [201]

We apply our classification alogrithm to the cancer image encodings generated from National Cancer Institute's GDC Portal [201]. The dataset consists of 1041 colon cancer images with 705 training and 336 testing images. The goal is to classify them to two classes, Colon Adenocarcinoma(CA) and Colon Mucinous Adenocarcinoma(CMA). American Cancer Society [202, 203] reports that colon cancer is responsible for one-thirds of all cancer related deaths in the country (about 101K cases and 51K deaths estimated in 2009). Use of Machine Learning Techniques with FCA is anticipated to enable faster processing and earlier detection of colateral cancer images and thereby reducing its overall risk and increasing the chances of patients survival.

The results are presented in Table 4.2. Our algorithm reported an overall accuracy of 86.84% with training accuracy of 82.84% and testing accuracy of 95.24%. When the attributes are scaled, the overall accuracy increased to 98.27%.

5.3.3 Discussion

LearnFCA has proven to be successful with varying degrees on the three datasets. The following general observations were made:

1. *Encoding Type (binary vs floating)*: Floating encodings represent data in a range compared to binary encodings (0 or 1) and hence are expected to capture features more efficiently. LearnFCA performs atleast as better on floating encodings as compared to the binary encodings in all the three datasets (*mnist* 30-100%, *omniglot* 2-55% , *cancer images* 86-98%).

2. *Scaling of Features*: Feature scaling which applies to only the floating encodings proved to be another mechanism to capture relevant features efficiently.

Dataset	Encoding Type	Total Objects	Classes	Train Size	Test Size	Scaled Attributes	Concepts	Train Accuracy	Test Accuracy	Overall Accuracy
mnist1	*binary*	500	10	400	100	64	2081	25.25	25.00	25.20
	floating	500	10	400	100	64	2081	25.50	23.00	25.00
						256	7996	36.75	41.00	37.60
						512	26019	96.00	98.00	96.40
mnist2	*binary*	100	10	80	20	64	2081	33.75	15.00	30.00
	floating	100	10	80	20	64	2081	31.25	20.00	29.00
						256	7088	93.75	100.00	95.00
						512	8249	100.00	100.00	100.00
omniglot (1000)	*floating*	1000	512	800	200	64	1832	2.50	1.50	2.30
	floating					256	6487	38.50	38.13	38.20
	floating					512	16411	54.50	59.00	55.40
omniglot (2000)	*floating*	2000	634	1600	400	64	1832	1.50	2.25	1.65
	floating					256	7091	24.31	21.00	23.65
	floating					512	18524	39.19	41.00	39.55
omniglot (3000)	*floating*	3000	652	2400	600	64	1832	1.75	0.83	1.56
	floating					256	7155	17.79	18.50	17.93
	floating					512	19994	32.75	35.00	33.2
omniglot (5000)	*floating*	5000	659	4000	1000	64	1833	1.15	0.90	1.10
	floating					256	7205	12.83	15.10	13.28
	floating					512	21870	25.58	25.80	25.62
cancer images	*floating*	1041	2	705	336	64	2077	82.84	95.24	86.84
	floating					256	6419	89.50	99.11	92.60
	floating					512	13164	97.59	99.70	98.27

Table 5.3: Results of applying LearnFCA onto various datasets

When scaling the features to four times, the accuracy of our model increased considerably to about 100% in all the three datasets (*mnist* 29-100%, *omniglot* 38-55% , *cancer images* 92-98%).

3. *Number of Classes*: Number of classes is an very important factor for classification accuracy. It is expected, that the more the number of classes, the harder it is to classify and lower the performance. With *omniglot* data which had the highest number of classes (about 650), the classification accuracy was just about 33%. With the *cancer images* (2 classes) , and *mnist* data (10 classes), our model performed close to 98% accurate.

4. *Size of Dataset*: The number of instances in the dataset plays a good role in the learning and classification process. With higher size of dataset, it is

expected that a model would learn and classify more efficiently and perform better. LearnFCA performs better with smaller size data (till 500), but its accuracy drops with very large datasets. Building the fuzzy lattice is exponential, the process is slow and hence the classification accuracy drops. With about 5000 images from *omniglot* dataset, the process took about 2 hours and classification accuracy was just about 25%. Our model was not able to run the entire *omniglot* dataset (1623 classes,14000 instances) and this stands for further analysis for future work.

5.4 Conclusion

This thesis provided a comprehensive review on FCA,its generalizations and their applications in various domains related to data mining and machine learning. We reviewed about 150 papers related to FCA and categorized them into three major areas, data analysis, knowledge management and learning, and classification. We also proposed LearnFCA, a classifier based on Fuzzy FCA and probabilistic techniques for learning, and classification purposes. Our proposed model was evaluated on three datasets with varying number of classes with varying degrees of success.

5.5 Future Work

The work presented in this thesis can be extended in several ways. Most of them follow from our analysis and our experimental results in previous section. Following are some research directions for future work:

One improvement to our LearnFCA is to explore techniques to efficiently build the concept lattice using incremental algorithms like the AddIntent [204], InClose2 [205] and [206, 207]. This would make processing faster and help in scaling to larger datasets. Other suggested approaches are noise removal before processing, ice-berging (removal of irrelevant concepts during lattice generation) [36], and use of context reduction techniques [208].

Use of FCA with big data and streaming data has not been investigated much until recently [7] and this is an emerging research area. There is a need to build efficient classifiers and tools to support various intelligent processes. One limitation of LearnFCA is its inability to scale to larger datasets and big data. Extending the model by using python packages, it's in-built, bench-marked tools for FCA and machine learning packages is another direction to explore, which might enable the model's usage with with big and streaming data.

Finally, LearnFCA could be combined with Temporal FCA, Ontology and RCA to enable our model have the ability to persist information, classification rules and save their changes over time. This would help us in retrieving snapshots of data at various time and even predicting changes in future.

Bibliography

[1] Jan Outrata. Preprocessing input data for machine learning by fca. In *CLA*, pages 187–198, 2010.

[2] Thomas Tilley and B Info Tech. *Formal concept analysis applications to requirements engineering and design*. University of Queensland, 2004.

[3] Sergei Obiedkov. *Introduction to Formal Concept Analysis*, 2019. https://www.coursera.org/learn/formal-concept-analysis.

[4] Jonas Poelmans, Dmitry I Ignatov, Sergei O Kuznetsov, and Guido Dedene. Formal concept analysis in knowledge processing: A survey on applications. *Expert systems with applications*, 40(16):6538–6560, 2013.

[5] Jonas Poelmans, Sergei O Kuznetsov, Dmitry I Ignatov, and Guido Dedene. Formal concept analysis in knowledge processing: A survey on models and techniques. *Expert systems with applications*, 40(16):6601–6623, 2013.

[6] Bernhard Ganter and Rudolf Wille. *Formal concept analysis: mathematical foundations*. Springer Science & Business Media, 2012.

[7] Carmen De Maio, Giuseppe Fenza, Vincenzo Loia, and Francesco Orciuoli. Distributed online temporal fuzzy concept analysis for stream processing in smart cities. *Journal of Parallel and Distributed Computing*, 110:31–41, 2017.

[8] Radim Belohlavek. Introduction to formal concept analysis. *Palacky University, Department of Computer Science, Olomouc*, 47, 2008.

[9] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[10] Siegfried Gottwald. An early approach toward graded identity and graded membership in set theory. *Fuzzy Sets and Systems*, 161(18):2369–2379, 2010.

[11] Ana Burusco Juandeaburre and Ramón Fuentes-González. The study of the l-fuzzy concept lattice. *Mathware & soft computing. 1994 Vol. 1 Núm. 3 p. 209-218*, 1994.

[12] R Belohlávek. Fuzzy concepts and conceptual structures: induced similarities. In *Joint Conference on Information Sciences*, volume 98, pages 179–182, 1998.

[13] Radim Belohlávek. Fuzzy closure operators. *Journal of mathematical analysis and applications*, 262(2):473–489, 2001.

[14] Radim Belohlávek, Tat'ána Funioková, and Vilém Vychodil. Fuzzy closure operators with truth stressers. *Logic Journal of the IGPL*, 13(5):503–513, 2005.

[15] Radim Bělohlávek and Vilém Vychodil. Attribute implications in a fuzzy setting. In *Formal Concept Analysis*, pages 45–60. Springer, 2006.

[16] Radim Belohlavek, Bernard De Baets, Jan Outrata, and Vilem Vychodil. Computing the lattice of all fixpoints of a fuzzy closure operator. *IEEE Transactions on Fuzzy Systems*, 18(3):546–557, 2010.

[17] Zdzislaw Pawlak. Rough set theory and its applications to data analysis. *Cybernetics & Systems*, 29(7):661–688, 1998.

[18] Marzena Kryszkiewicz. Rough set approach to incomplete information systems. *Information sciences*, 112(1-4):39–49, 1998.

[19] Robert E Kent. Rough concept analysis: a synthesis of rough sets and formal concept analysis. *Fundamenta informaticae*, 27(2, 3):169–181, 1996.

[20] Yiyu Yao and Yaohua Chen. Subsystem based generalizations of rough set approximations. In *International Symposium on Methodologies for Intelligent Systems*, pages 210–218. Springer, 2005.

[21] Qinghua Zhang, Qin Xie, and Guoyin Wang. A survey on rough set theory and its applications. *CAAI Transactions on Intelligence Technology*, 1(4):323–333, 2016.

[22] Jitender S Deogun and Jamil Saquer. Monotone concepts for formal concept analysis. *Discrete Applied Mathematics*, 144(1-2):70–78, 2004.

[23] Jamil Saquer and Jitender S Deogun. Formal rough concept analysis. In *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, pages 91–99. Springer, 1999.

[24] Alexander Demin, Denis Ponomaryov, and Evgeny Vityaev. Probabilistic concepts in formal contexts. In *International Andrei Ershov Memorial Conference on Perspectives of System Informatics*, pages 394–410. Springer, 2011.

[25] Francesco Kriegel. Probabilistic implication bases in fca and probabilistic bases of gcis in. *International Journal of General Systems*, 46(5):511–546, 2017.

[26] EE Vityaev, Alexander V Demin, and Denis K Ponomaryov. Probabilistic generalization of formal concepts. *Programming and Computer Software*, 38(5):219–230, 2012.

[27] Jitender Deogun, Liying Jiang, Ying Xie, and Vijay Raghavan. Probability logic modeling of knowledge discovery in databases. In *International Symposium on Methodologies for Intelligent Systems*, pages 402–407. Springer, 2003.

[28] Karl Erich Wolff. Concepts, states, and systems. In *AIP Conference Proceedings*, volume 517, pages 83–97. AIP, 2000.

[29] Sébastien Ferré and Olivier Ridoux. A logical generalizationof formal concept analysis. In *International Conference on Conceptual Structures*, pages 371–384. Springer, 2000.

[30] Susanne Prediger. Logical scaling in formal concept analysis. In *International Conference on Conceptual Structures*, pages 332–341. Springer, 1997.

[31] Sébastien Ferré. Negation, opposition, and possibility in logical concept analysis. In *Formal Concept Analysis*, pages 130–145. Springer, 2006.

[32] Yan Zhao and Yiyu Yao. Classification based on logical concept analysis. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 419–430. Springer, 2006.

[33] Mohamed Hacene Rouane, Marianne Huchard, Amedeo Napoli, and Petko Valtchev. A proposal for combining formal concept analysis and description logics for mining relational data. In *International Conference on Formal Concept Analysis*, pages 51–65. Springer, 2007.

[34] Mohamed Rouane-Hacene, Marianne Huchard, Amedeo Napoli, and Petko Valtchev. Relational concept analysis: mining concept lattices from multi-relational data. *Annals of Mathematics and Artificial Intelligence*, 67(1):81–108, 2013.

[35] David Grosser. *Galois lattice interactive constructor*, 2019.

[36] Alexandre Bazin, Jessie Carbonnel, Marianne Huchard, Giacomo Kahn, Priscilla Keip, and Amirouche Ouzerdine. On-demand relational concept analysis. In *International Conference on Formal Concept Analysis*, pages 155–172. Springer, 2019.

[37] Fritz Lehmann and Rudolf Wille. A triadic approach to formal concept analysis. In *International Conference on Conceptual Structures*, pages 32–43. Springer, 1995.

[38] Jan Konecny and Petr Osicka. General approach to triadic concept analysis. In *CLA*, pages 116–126, 2010.

[39] Radim Belohlavek and Petr Osicka. Triadic concept analysis of data with fuzzy attributes. In *2010 IEEE International Conference on Granular Computing*, pages 661–665. IEEE, 2010.

[40] Radim Belohlavek and Petr Osicka. Triadic concept lattices of data with graded attributes. *International Journal of General Systems*, 41(2):93–108, 2012.

[41] Dmitry I Ignatov, Dmitry V Gnatyshak, Sergei O Kuznetsov, and Boris G Mirkin. Triadic formal concept analysis and triclustering: searching for optimal patterns. *Machine Learning*, 101(1-3):271–302, 2015.

[42] Sergei O Kuznetsov. Pattern structures for analyzing complex data. In *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, pages 33–44. Springer, 2009.

[43] SO Kuznetsov. Jsm-method as a machine learning method. *Itogi Nauki i Tekhniki, ser. Informatika*, 15:17–50, 1991.

[44] Sergei O Kuznetsov. Learning of simple conceptual graphs from positive and negative examples. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 384–391. Springer, 1999.

[45] Mehdi Kaytoue, Sébastien Duplessis, Sergei O Kuznetsov, and Amedeo Napoli. Two fca-based methods for mining gene expression data. In *International Conference on Formal Concept Analysis*, pages 251–266. Springer, 2009.

[46] Zainab Assaghir, Mehdi Kaytoue, Wagner Meira, and Jean Villerd. Extracting decision trees from interval pattern concept lattices. In *International Conference on Concept Lattices and their Applications-CLA*, volume 2011, 2011.

[47] Lotfi Asker Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1(1):3–28, 1978.

[48] Didier Dubois and Henri Prade. Possibility theory and formal concept analysis: Characterizing independent sub-contexts. *Fuzzy Sets and Systems*, 196:4–16, 2012.

[49] Yassine Djouadi, Didier Dubois, and Henri Prade. Possibility theory and formal concept analysis: Context decomposition and uncertainty handling. In *International conference on information processing and management of uncertainty in knowledge-based systems*, pages 260–269. Springer, 2010.

[50] Yassine Djouadi and Henri Prade. Possibility-theoretic extension of derivation operators in formal concept analysis over fuzzy lattices. *Fuzzy Optimization and Decision Making*, 10(4):287, 2011.

[51] Linton C Freeman. Cliques, galois lattices, and the structure of human social groups. *Social networks*, 18(3):173–187, 1996.

[52] Nicolas Maille, Irving C Statler, and Laurent Chaudron. An application of fca to the analysis of aeronautical incidents. In *International Conference on Formal Concept Analysis*, pages 145–161. Springer, 2005.

[53] Nyeint Nyeint Myat and Khin Haymar Saw Hla. A combined approach of formal concept analysis and text mining for concept based document clustering. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 330–333. IEEE Computer Society, 2005.

[54] Yoshiaki Okubo and Makoto Haraguchi. Finding conceptual document clusters with improved top-n formal concept search. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 347–351. IEEE Computer Society, 2006.

[55] Sergei Kuznetsov, Sergei Obiedkov, and Camille Roth. Reducing the representation complexity of lattice-based taxonomies. In *International Conference on Conceptual Structures*, pages 241–254. Springer, 2007.

[56] Camille Roth, Sergei Obiedkov, and Derrick Kourie. Towards concise representation for taxonomies of epistemic communities. In *International Conference on Concept Lattices and Their Applications*, pages 240–255. Springer, 2006.

[57] Camille Roth, Sergei Obiedkov, and Derrick G Kourie. On succinct representation of knowledge community taxonomies with formal concept analysis. *International Journal of Foundations of Computer Science*, 19(02):383–404, 2008.

[58] Thomas Eisenbarth, Rainer Koschke, and Daniel Simon. Locating features in source code. *IEEE Transactions on software engineering*, 29(3):210–224, 2003.

[59] Paolo Tonella and Mariano Ceccato. Aspect mining through the formal concept analysis of execution traces. In *11th Working Conference on Reverse Engineering*, pages 112–121. IEEE, 2004.

[60] Peter Busch and Debbie Richards. Modelling tacit knowledge via questionnaire data. In *International Conference on Formal Concept Analysis*, pages 321–328. Springer, 2004.

[61] Anna Formica. Ontology-based concept similarity in formal concept analysis. *Information sciences*, 176(18):2624–2641, 2006.

[62] Anna Formica. Concept similarity in formal concept analysis: An information content approach. *Knowledge-Based Systems*, 21(1):80–87, 2008.

[63] Thomas Girault. Concept lattice mining for unsupervised named entity annotation. In *Proc. CLA*, pages 35–46. Citeseer, 2008.

[64] Eugen Lounkine, Jens Auer, and Jürgen Bajorath. Formal concept analysis for the identification of molecular fragment combinations specific for active and highly potent compounds. *Journal of medicinal chemistry*, 51(17):5342–5348, 2008.

[65] Mohamed Rouane-Hacene, Yannick Toussaint, and Petko Valtchev. Mining safety signals in spontaneous reports database using concept analysis. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 285–294. Springer, 2009.

[66] Jean Villerd, Yannick Toussaint, and Agnès Lillo-Le Louët. Adverse drug reaction mining in pharmacovigilance data using formal concept analysis. In

Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 386–401. Springer, 2010.

[67] Martin Ebner, Herbert Mühlburger, Sandra Schaffert, Mandy Schiefner, Wolfgang Reinhardt, and Steve Wheeler. Getting granular on twitter: tweets from a conference and their limited usefulness for non-participants. In *IFIP International Conference on Key Competencies in the Knowledge Society*, pages 102–113. Springer, 2010.

[68] Jonas Poelmans, Guido Dedene, Gerda Verheyden, Herman Van Der Mussele, Stijn Viaene, and Edward Peters. Combining business process and data discovery techniques for analyzing and improving integrated care pathways. In *Industrial Conference on Data Mining*, pages 505–517. Springer, 2010.

[69] Dagmar Stumpfe, Eugen Lounkine, and Jürgen Bajorath. Molecular test systems for computational selectivity studies and systematic analysis of compound selectivity profiles. In *Chemoinformatics and computational chemical biology*, pages 503–515. Springer, 2011.

[70] Elias Egho, Nicolas Jay, Chedy Raïssi, and Amedeo Napoli. A fca-based analysis of sequential care trajectories. In *The Eighth International Conference on Concept Lattices and their Applications-CLA 2011*, 2011.

[71] Dominik Endres, Ruth Adam, Martin A Giese, and Uta Noppeney. Understanding the semantic structure of human fmri brain recordings with formal concept analysis. In *International Conference on Formal Concept Analysis*, pages 96–111. Springer, 2012.

[72] Paul Elzinga, Karl Erich Wolff, and Jonas Poelmans. Analyzing chat conversations of pedophiles with temporal relational semantic systems. In *2012 European Intelligence and Security Informatics Conference*, pages 242–249. IEEE, 2012.

[73] Jérémy Besson, Céline Robardet, and Jean-François Boulicaut. Constraint-based mining of formal concepts in transactional data. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 615–624. Springer, 2004.

[74] Kim Mens and Tom Tourwé. Delving source code with formal concept analysis. *Computer Languages, Systems & Structures*, 31(3-4):183–197, 2005.

[75] Vladimır Sklenár, Jirı Zacpal, and Erik Sigmund. Evaluation of ipaq questionnaire by fca. In *CLA*, pages 60–69, 2005.

[76] L John Old. Homograph disambiguation using formal concept analysis. In *Formal Concept Analysis*, pages 221–232. Springer, 2006.

[77] Silvia Breu, Thomas Zimmermann, and Christian Lindig. Mining eclipse for cross-cutting concerns. In *Proceedings of the 2006 international workshop on Mining software repositories*, pages 94–97. ACM, 2006.

[78] K Fang, C Chang, and Y Chi. Using formal concept analysis to leverage ontology-based acu-point knowledge system. d. zhang (ed.): Icmb, lncs 4901, 115-121, 2007.

[79] Ghassan Beydoun, Roman Kultchitsky, and Grace Manasseh. Evolving semantic web with social navigation. *Expert Systems with Applications*, 32(2):265–276, 2007.

[80] Concettina Del Grosso, Massimiliano Di Penta, and Ignacio Garcia-Rodriguez de Guzman. An approach for mining services in database oriented applications. In *11th European Conference on Software Maintenance and Reengineering (CSMR'07)*, pages 287–296. IEEE, 2007.

[81] Radim Belohlavek. A note on variable threshold concept lattices: threshold-based operators are reducible to classical concept-forming operators. *Information Sciences*, 177(15):3186–3191, 2007.

[82] Radim Belohlavek, Erik Sigmund, and Jiří Zacpal. Evaluation of ipaq questionnaires supported by formal concept analysis. *Information Sciences*, 181(10):1774–1786, 2011.

[83] Kenji Sato, Yoshiaki Okubo, Makoto Haraguchi, and Susumu Kunifuji. Data mining of time-series medical data by formal concept analysis. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 1214–1221. Springer, 2007.

[84] Yi Zhao, Wolfgang Halang, and Xia Wang. Rough ontology mapping in e-business integration. In *E-Service Intelligence*, pages 75–93. Springer, 2007.

[85] Hak Lae Kim, Suk Hyung Hwang, and Hong Gee Kim. Fca-based approach for mining contextualized folksonomy. In *Proceedings of the 2007 ACM symposium on Applied computing*, pages 1340–1345. ACM, 2007.

[86] Yao Lu and Guoyi Zhang. A dl-based approach for eliminating concept restriction conflicts in ontology merging. In *2008 IFIP International Conference on Network and Parallel Computing*, pages 446–451. IEEE, 2008.

[87] Peggy Cellier, Mireille Ducassé, Sébastien Ferré, and Olivier Ridoux. Formal concept analysis enhances fault localization in software. In *International Conference on Formal Concept Analysis*, pages 273–288. Springer, 2008.

[88] Ian Molloy, Hong Chen, Tiancheng Li, Qihua Wang, Ninghui Li, Elisa Bertino, Seraphin Calo, and Jorge Lobo. Mining roles with semantic meanings. In *Proceedings of the 13th ACM symposium on Access control models and technologies*, pages 21–30. ACM, 2008.

[89] Susanne Motameny, Beatrix Versmold, and Rita Schmutzler. Formal concept analysis for the identification of combinatorial biomarkers in breast cancer. In *International Conference on Formal Concept Analysis*, pages 229–240. Springer, 2008.

[90] Jutta Gebert, Susanne Motameny, Ulrich Faigle, Christian V Forst, and Rainer Schrader. Identifying genes of gene regulatory networks using formal concept analysis. *Journal of Computational Biology*, 15(2):185–194, 2008.

[91] Giuseppe Fenza, Vincenzo Loia, and Sabrina Senatore. Concept mining of semantic web services by means of extended fuzzy formal concept analysis (ffca). In *2008 IEEE International Conference on Systems, Man and Cybernetics*, pages 240–245. IEEE, 2008.

[92] Giuseppe Fenza and Sabrina Senatore. Friendly web services selection exploiting fuzzy formal concept analysis. *Soft Computing*, 14(8):811–819, 2010.

[93] Aurélie Bertaux, Florence Le Ber, Agnès Braud, and Michèle Trémolières. Identifying ecological traits: a concrete fca-based approach. In *International Conference on Formal Concept Analysis*, pages 224–236. Springer, 2009.

[94] Valmi Dufour-Lussier, Jean Lieber, Emmanuel Nauer, and Yannick Toussaint. Text adaptation using formal concept analysis. In *International Conference on Case-Based Reasoning*, pages 96–110. Springer, 2010.

[95] Jonas Poelmans, Paul Elzinga, Stijn Viaene, and Guido Dedene. A method based on temporal concept analysis for detecting and profiling human trafficking suspects. In *Artificial Intelligence and Applications*, pages 1–9. Acta Press, 2010.

[96] Jonas Poelmans, Paul Elzinga, Guido Dedene, Stijn Viaene, and Sergei O Kuznetsov. A concept discovery approach for fighting human trafficking and forced prostitution. In *International Conference on Conceptual Structures*, pages 201–214. Springer, 2011.

[97] Jonas Poelmans, Dmitry I Ignatov, Stijn Viaene, Guido Dedene, and Sergei O Kuznetsov. Text mining scientific papers: a survey on fca-based information retrieval research. In *Industrial Conference on Data Mining*, pages 273–287. Springer, 2012.

[98] Paul Elzinga, Jonas Poelmans, Stijn Viaene, Guido Dedene, and Shanti Morsing. Terrorist threat assessment with formal concept analysis. In *2010 IEEE International Conference on Intelligence and Security Informatics*, pages 77–82. IEEE, 2010.

[99] Ingrid Falk, Claire Gardent, and Alejandra Lorenzo. Using formal concept analysis to acquire knowledge about verbs. In *7th International Conference on Concept Lattices and Their Applications-CLA 2010*, page 12, 2010.

[100] Ingrid Falk and Claire Gardent. Combining formal concept analysis and translation to assign frames and semantic role sets to french verbs. *Annals of Mathematics and Artificial Intelligence*, 70(1-2):123–150, 2014.

[101] Nizar Messai, Jacques Bouaud, Marie-Aude Aufaure, Laurent Zelek, and Brigitte Séroussi. Using formal concept analysis to discover patterns of non-compliance with clinical practice guidelines: a case study in the management of breast cancer. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 119–128. Springer, 2011.

[102] Benjamin J Keller, Felix Eichinger, and Matthias Kretzler. Formal concept analysis of disease similarity. *AMIA Summits on Translational Science Proceedings*, 2012:42, 2012.

[103] Uta Priss, Peter Riegler, and Nils Jensen. Using fca for modelling conceptual difficulties in learning processes. In *Contributions to the 10th International Conference on Formal Concept Analysis (ICFCA 2012)*, pages 161–173, 2012.

[104] Woo-Chul Cho and Debbie Richards. Improvement of precision and recall for information retrieval in a narrow domain: reuse of concepts by formal concept analysis. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)*, pages 370–376. IEEE, 2004.

[105] Sebastian Rudolph. Exploring relational structures via fle. In *International Conference on Conceptual Structures*, pages 196–212. Springer, 2004.

[106] Baoyao Zhou, Siu Cheung Hui, and Kuiyu Chang. A formal concept analysis approach for web usage mining. In *International Conference on Intelligent Information Processing*, pages 437–441. Springer, 2004.

[107] Brandon M Hauff and Jitender S Deogun. Parameter tuning for disjoint clusters based on concept lattices with application to location learning. In *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, pages 232–239. Springer, 2007.

[108] Simon Colton and Daniel Wagner. Using formal concept analysis in mathematical discovery. In *Towards Mechanized Mathematical Assistants*, pages 205–220. Springer, 2007.

[109] Dmitry I Ignatov and Sergei O Kuznetsov. Concept-based recommendations for internet advertisement. *arXiv preprint arXiv:0906.4982*, 2009.

[110] Thanh Tho Quan, Siu Cheung Hui, and Alvis Cheuk M Fong. Automatic fuzzy ontology generation for semantic help-desk support. *IEEE Transactions on Industrial Informatics*, 2(3):155–164, 2006.

[111] Nicolas Jay, François Kohler, and Amedeo Napoli. Analysis of social communities with iceberg and stability-based concept lattices. In *International conference on formal concept analysis*, pages 258–272. Springer, 2008.

[112] Dongmei Huang and Hui Zhu. Rough formal concept analysis for semi-automatic marine ontology. In *2008 Fourth International Conference on Natural Computation*, volume 5, pages 114–118. IEEE, 2008.

[113] Yingyu Wang, Yajun Du, and Shaoming Chen. The understanding between two agent crawlers based on domain ontology. In *2009 International Conference on Computational Intelligence and Natural Computing*, volume 1, pages 47–50. IEEE, 2009.

[114] Frithjof Dau and Martin Knechtel. Access policy design supported by fca methods. In *International Conference on Conceptual Structures*, pages 141–154. Springer, 2009.

[115] Marina Z Solesvik and Sylvia Encheva. Partner selection for interfirm collaboration: the context of ship design. In *International Conference on Cooperative Design, Visualization and Engineering*, pages 141–144. Springer, 2009.

[116] Dmitry Ignatov, Serafima Mamedova, Nikita Romashkin, and Ivan Shamshurin. What can closed sets of students and their marks say? pages 223–228, 2011.

[117] Nikita Romashkin, Dmitry Ignatov, and Elena Kolotova. How university entrants are choosing their department? mining of university admission process with fca taxonomies. In *Educational Data Mining 2011*, 2010.

[118] Richard Cole and Gerd Stumme. Cem–a conceptual email manager. In *International Conference on Conceptual Structures*, pages 438–452. Springer, 2000.

[119] Richard Cole and Peter Eklund. Browsing semi-structured web texts using formal concept analysis. In *International Conference on Conceptual Structures*, pages 319–332. Springer, 2001.

[120] Richard Cole, Peter Eklund, and Gerd Stumme. Document retrieval for e-mail search and discovery using formal concept analysis. *Applied artificial intelligence*, 17(3):257–280, 2003.

[121] Peter Eklund, Jon Ducrou, and Peter Brawn. Concept lattices for information visualization: Can novices read line-diagrams? In *International Conference on Formal Concept Analysis*, pages 57–73. Springer, 2004.

[122] Gerd Stumme and Alexander Maedche. Fca-merge: Bottom-up merging of ontologies. In *IJCAI*, volume 1, pages 225–230, 2001.

[123] Olivier Curé and Robert Jeansoulin. An fca-based solution for ontology mediation. In *Proceedings of the 2nd international workshop on Ontologies and information systems for the semantic web*, pages 39–46. ACM, 2008.

[124] Claudio Carpineto, Giovanni Romano, and Fondazione Ugo Bordoni. Exploiting the potential of concept lattices for information retrieval with credo. *J. UCS*, 10(8):985–1013, 2004.

[125] Kleber Xavier Sampaio de Souza and Joseph Davis. Aligning ontologies and evaluating concept similarities. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 1012–1029. Springer, 2004.

[126] Kleber Xavier Sampaio de Souza and Joseph Davis. Using an aligned ontology to process user queries. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pages 44–53. Springer, 2004.

[127] Debbie Richards. Addressing the ontology acquisition bottleneck through reverse ontological engineering. *Knowledge and Information Systems*, 6(4):402–427, 2004.

[128] Richard Cole, Thomas Tilley, and Jon Ducrou. Conceptual exploration of software structure: A collection of examples. In *CLA*, pages 135–148, 2005.

[129] Bjoern Koester et al. Conceptual knowledge processing with google. In *LWA*, pages 178–183, 2005.

[130] Bjoern Koester. Conceptual knowledge retrieval with fooca: Improving web search engine results with contexts and concept hierarchies. In *Industrial Conference on Data Mining*, pages 176–190. Springer, 2006.

[131] Frithjof Dau, Jon Ducrou, and Peter Eklund. Concept similarity and related categories in searchsleuth. In *International Conference on Conceptual Structures*, pages 255–268. Springer, 2008.

[132] Jon Ducrou and Peter Eklund. Searchsleuth: The conceptual neighbourhood of an web query. pages 249—260, 2007.

[133] Yuekui Yang, Yajun Du, Jingyu Sun, and Yufeng Hai. A topic-specific web crawler with concept similarity context graph based on fca. In *International Conference on Intelligent Computing*, pages 840–847. Springer, 2008.

[134] YaJun Du and HaiMing Li. Strategy for mining association rules for web pages based on formal concept analysis. *Applied Soft Computing*, 10(3):772–783, 2010.

[135] Alessandro Marco Boutari, Claudio Carpineto, and Raffaele Nicolussi. Evaluating term concept association mesaures for short text expansion: Two case studies of classification and clustering. In *CLA*, pages 163–174, 2010.

[136] Uta Priss and L John Old. Modelling lexical databases with formal concept analysis. *J. UCS*, 10(8):967–984, 2004.

[137] Uta Priss and L John Old. Concept neighbourhoods in lexical databases. In *International Conference on Formal Concept Analysis*, pages 283–295. Springer, 2010.

[138] Dong-Soon Kim, Suk-Hyung Hwang, and Hong-Gee Kim. Concept analysis of owl ontology based on the context family model. In *2007 International Conference on Convergence Information Technology (ICCIT 2007)*, pages 896–901. IEEE, 2007.

[139] Michel Wermelinger, Yijun Yu, and Markus Strohmaier. Using formal concept analysis to construct and visualise hierarchies of socio-technical relations. In *2009 31st International Conference on Software Engineering-Companion Volume*, pages 327–330. IEEE, 2009.

[140] V Choi, Y HUANG, V LAM, D POTTER, R LAUBENBACHER, and K DUCA. Using formal concept analysis for microarray data comparison. In *Proceedings Of The 5th Asia-Pacific Bioinformatics Conference*, pages 57–66. World Scientific, 2007.

[141] Vicky Choi, Yang Huang, Vy Lam, Dustin Potter, Reinhard Laubenbacher, and Karen Duca. Using formal concept analysis for microarray data comparison. *Journal of bioinformatics and computational biology*, 6(01):65–75, 2008.

[142] Telung Pan and Kwoting Fang. Ontology-based formal concept differences analysis in radiology report impact by the adoption of pacs. In *International Conference on Formal Concept Analysis*, pages 267–283. Springer, 2009.

[143] Bénédicte Le Grand, Michel Soto, and Marie-Aude Aufaure. Conceptual and spatial footprints for complex systems analysis: Application to the semantic web. In *International Conference on Database and Expert Systems Applications*, pages 114–127. Springer, 2009.

[144] Guoqian Jiang, Jyotishman Pathak, and Christopher G Chute. Formalizing icd coding rules using formal concept analysis. *Journal of Biomedical Informatics*, 42(3):504–517, 2009.

[145] Barış Sertkaya. Ontocomp: A protege plugin for completing owl ontologies. In *European Semantic Web Conference*, pages 898–902. Springer, 2009.

[146] Guoqian Jiang and Christopher G Chute. Auditing the semantic completeness of snomed ct using formal concept analysis. *Journal of the American Medical Informatics Association*, 16(1):89–102, 2009.

[147] Mehdi Kaytoue, Sergei O Kuznetsov, Juraj Macko, Wagner Meira, and Amedeo Napoli. Mining biclusters of similar values with triadic concept analysis. *arXiv preprint arXiv:1111.3270*, 2011.

[148] Markus Kirchberg, Erwin Leonardi, Yu Shyang Tan, Sebastian Link, Ryan KL Ko, and Bu Sung Lee. Formal concept discovery in semantic web data. In *International Conference on Formal Concept Analysis*, pages 164–179. Springer, 2012.

[149] Guoqian Jiang, Katsuhiko Ogasawara, Akira Endoh, and Tsunetaro Sakurai. Context-based ontology building support in clinical domains using formal concept analysis. *International journal of medical informatics*, 71(1):71–81, 2003.

[150] Keanhuat Soon and Werner Kuhn. Formalizing user actions for ontologies. In *International Conference on Geographic Information Science*, pages 299–312. Springer, 2004.

[151] Philipp Cimiano, Andreas Hotho, Gerd Stumme, and Julien Tane. Conceptual knowledge processing with formal concept analysis and ontologies. In *International Conference on Formal Concept Analysis*, pages 189–207. Springer, 2004.

[152] Yu-Liang Chi, Tien-Yu Hsu, and Wei-Pang Yang. Building ontological knowledge bases for sharing knowledge in digital archive. In *2005 International Conference on Machine Learning and Cybernetics*, volume 4, pages 2261–2266. IEEE, 2005.

[153] Suk-Hyung Hwang, Hong-Gee Kim, and Hae-Sool Yang. A fca-based ontology construction for the design of class hierarchy. In *International Conference on Computational Science and Its Applications*, pages 827–835. Springer, 2005.

[154] Wei Xu, Wenjie Li, Mingli Wu, Wei Li, and Chunfa Yuan. Deriving event relevance from the ontology constructed with formal concept analysis. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 480–489. Springer, 2006.

[155] Debbie Richards. Ad-hoc and personal ontologies: A prototyping approach to ontology engineering. In *Pacific Rim Knowledge Acquisition Workshop*, pages 13–24. Springer, 2006.

[156] Quan Thanh Tho, Siu Cheung Hui, Alvis Cheuk M Fong, and Tru Hoang Cao. Automatic fuzzy ontology generation for semantic web. *IEEE transactions on knowledge and data engineering*, 18(6):842–856, 2006.

[157] Johannes Wollbold, Reinhard Guthke, and Bernhard Ganter. Constructing a knowledge base for gene regulatory dynamics by formal concept analysis methods. In *International Conference on Algebraic Biology*, pages 230–244. Springer, 2008.

[158] Johannes Wollbold, René Huber, Dirk Pohlers, Dirk Koczan, Reinhard Guthke, Raimund W Kinne, and Ulrike Gausmann. Adapted boolean network models for extracellular matrix formation. *BMC systems biology*, 3(1):77, 2009.

[159] Hui Xu and Debao Xiao. Building information specification ontology for computer network management based on formal concept analysis. In *2009 International Conference on Information and Automation*, pages 312–317. IEEE, 2009.

[160] Debbie Richards and Usama Malik. Multi-level knowledge discovery from rule bases. *Applied Artificial Intelligence*, 17(3):181–205, 2003.

[161] Philipp Cimiano, Andreas Hotho, and Steffen Staab. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of artificial intelligence research*, 24:305–339, 2005.

[162] Pablo Gamallo, Gabriel P Lopes, and Alexandre Agustini. Inducing classes of terms from text. In *International Conference on Text, Speech and Dialogue*, pages 31–38. Springer, 2007.

[163] Ching-Chieh Kiu and Chien-Sing Lee. Ontological knowledge management through hybrid unsupervised clustering techniques. In *Asia-Pacific Web Conference*, pages 499–510. Springer, 2008.

[164] Rokia Bendaoud, Yannick Toussaint, and Amedeo Napoli. Pactole: A methodology and a system for semi-automatically enriching an ontology from a collection of texts. In *International Conference on Conceptual Structures*, pages 203–216. Springer, 2008.

[165] Jean-Gabriel Ganascia. Charade: A rule system learning system. In *IJCAI*, volume 87, pages 234–239, 1987.

[166] Claudio Carpineto and Giovanni Romano. Galois: An order-theoretic approach to conceptual clustering. In *Proceedings of ICML*, volume 293, pages 33–40, 1993.

[167] Michel Liquière and E Mephu Nguifo. Legal: learning with galois lattice. *Actes des Journ ees Fran caises sur l'Apprentissage (JFA), Lannion, France*, pages 93–113, 1990.

[168] G Oosthuizen. The application of concept lattices to machine learning. *Technical Report CSTR 94/01, Pretoria, South Africa*, 1994.

[169] Mehran Sahami. Learning classification rules using lattices. In *European Conference on Machine Learning*, pages 343–346. Springer, 1995.

[170] P Njiwoua and E Mephu Nguifo. Améliorer l'apprentissage à partir d'instances grâce à l'induction de concepts: le système cible. *Revue d'intelligence artificielle*, 13(2):413–440, 1999.

[171] Zhipeng Xie, Wynne Hsu, Zongtian Liu, and Mong Li Lee. Concept lattice based composite classifiers for high predictability. *Journal of Experimental & Theoretical Artificial Intelligence*, 14(2-3):143–156, 2002.

[172] Bernhard Ganter and Sergei O Kuznetsov. Hypotheses and version spaces. In *International Conference on Conceptual Structures*, pages 83–95. Springer, 2003.

[173] Bernhard Ganter and Sergei O Kuznetsov. Formalizing hypotheses with concepts. In *International Conference on Conceptual Structures*, pages 342–356. Springer, 2000.

[174] Sergei O Kuznetsov. Complexity of learning in concept lattices from positive and negative examples. *Discrete Applied Mathematics*, 142(1-3):111–125, 2004.

[175] Sergei O Kuznetsov. Machine learning and formal concept analysis. In *International Conference on Formal Concept Analysis*, pages 287–312. Springer, 2004.

[176] Bernhard Ganter, Peter A Grigoriev, Sergei O Kuznetsov, and Mikhail V Samokhin. Concept-based data mining with scaled labeled graphs. In *International Conference on Conceptual Structures*, pages 94–108. Springer, 2004.

[177] Sergei O Kuznetsov and Mikhail V Samokhin. Learning closed sets of labeled graphs for chemical applications. In *International Conference on Inductive Logic Programming*, pages 190–208. Springer, 2005.

[178] Mohamed Aoun-Allah and Guy Mineau. Rule validation of a meta-classifier through a galois (concept) lattice and complementary means. In *International Conference on Concept Lattices and Their Applications*, pages 123–138. Springer, 2006.

[179] Marc Ricordeau. Q-concept-learning: generalization with concept lattice representation in reinforcement learning. In *Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence*, pages 316–323. IEEE, 2003.

[180] Marc Ricordeau and Michel Liquiere. Policies generalization in reinforcement learning using galois partitions lattices. In *CLA*, 2007.

[181] Sebastian Rudolph. Using fca for encoding closure operators into neural networks. In *International Conference on Conceptual Structures*, pages 321–332. Springer, 2007.

[182] ZHOU Wen, Zong-tian Liu, and ZHAO Yan. Ontology learning by clustering based on fuzzy formal concept analysis. In *31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*, volume 1, pages 204–210. IEEE, 2007.

[183] Yi-Hsing Chang and Hsiu-Yi Huang. An automatic document classifier system based on naive bayes classifier and ontology. In *2008 International Conference on Machine Learning and Cybernetics*, volume 6, pages 3144–3149. IEEE, 2008.

[184] Norbert Tsopzé, Engelbert Mephu Nguifo, and Gilbert Tindo. Clann: Concept lattice-based artificial neural network for supervised classification. In *CLA*, volume 331. Citeseer, 2007.

[185] Claudio Carpineto, Carla Michini, and Raffaele Nicolussi. A concept lattice-based kernel for svm text classification. In *International Conference on Formal Concept Analysis*, pages 237–250. Springer, 2009.

[186] Radim Belohlavek, Bernard De Baets, Jan Outrata, and Vilem Vychodil. Inducing decision trees via concept lattices. *International journal of general systems*, 38(4):455–467, 2009.

[187] Christian Lindig. Fast concept analysis. *Working with Conceptual Structures-Contributions to ICCS*, 2000:152–161, 2000.

[188] Muriel Visani, Karell Bertet, and J-M Ogier. Navigala: an original symbol classifier based on navigation through a galois lattice. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(04):449–473, 2011.

[189] Sergei O Kuznetsov. Fitting pattern structures to knowledge discovery in big data. In *International conference on formal concept analysis*, pages 254–266. Springer, 2013.

[190] Nida Meddouri, Hela Khoufi, and Mondher Maddouri. Parallel learning and classification for rules based on formal concepts. *Procedia Computer Science*, 35:358–367, 2014.

[191] Yury Kashnitsky and Dmitry I Ignatov. Can fca-based recommender system suggest a proper classifier? *arXiv preprint arXiv:1504.05473*, 2015.

[192] Mohamed Abdalhakem Taha Ali. Bagged randomized conceptual machine learning method. Master's thesis, 2018.

[193] Haithem Mezni and Taher Abdeljaoued. A cloud services recommendation system based on fuzzy formal concept analysis. *Data & Knowledge Engineering*, 116:100–123, 2018.

[194] Rawia Fray, Nida Meddouri, and Mondher Maddouri. Cloud implementation of classifier nominal concepts using distributedwekaspark. 2019.

[195] Gerd Stumme, Rafik Taouil, Yves Bastide, Nicolas Pasquier, and Lotfi Lakhal. Computing iceberg concept lattices with titanic. *Data & knowledge engineering*, 42(2):189–222, 2002.

[196] Sergei O Kuznetsov. On stability of a formal concept. *Annals of Mathematics and Artificial Intelligence*, 49(1):101–115, 2007.

[197] Sergei Objedkov Julien Tane, Tim Kaiser. *Conexp: The Concept Explorer*, 2006. http://conexp.sourceforge.net/.

[198] Eugen Kiss David Bormann, Jan Kassel. *ConExp-NG: A GUI-centric tool for Formal Concept Analysis*, 2013. https://github.com/fcatools/conexp-ng.

[199] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[200] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[201] National Cancer Institute. *National Cancer Institute, GDC Data Portal*, 2019. https://portal.gdc.cancer.gov/projects/TCGA-COAD.

[202] American Cancer Society. *Colorectal Cancer Facts & Figures 2017-2019*, 2019. https://www.cancer.org/content/dam/cancer-org/research/cancer-facts-and-statistics/colorectal-cancer-facts-and-figures/colorectal-cancer-facts-and-figures-2017-2019.pdf.

[203] American Cancer Society. *Cancer Facts & Figures*, 2019. https://www.cancer.org/content/dam/cancer-org/research/cancer-facts-and-statistics/annual-cancer-facts-and-figures/2019/cancer-facts-and-figures-2019.pdf.

[204] Dean Van Der Merwe, Sergei Obiedkov, and Derrick Kourie. Addintent: A new incremental algorithm for constructing concept lattices. In *International Conference on Formal Concept Analysis*, pages 372–385. Springer, 2004.

[205] Simon Andrews. In-close2, a high performance formal concept miner. In *International Conference on Conceptual Structures*, pages 50–62. Springer, 2011.

[206] Jingpu Zhang, Ronghui Liu, Ligeng Zou, and Licheng Zeng. A new rapid incremental algorithm for constructing concept lattices. *Information*, 10(2):78, 2019.

[207] Xenia Alexandre Naidenova. Incremental approach to classification learning.

In *Encyclopedia of Information Science and Technology, Fourth Edition*, pages 191–201. IGI Global, 2018.

[208] László Kovács. An algorithm using context reduction for efficient incremental generation of concept set. *Fundamenta Informaticae*, 165(1):43–73, 2019.