Formal Concept Analysis

The Galactic Organization <contact@thegalactic.org>

2018-2023 [1]

[1] © 2018-2023 the Galactic Organization. This document is licensed under CC-by-nc-nd (https://creativecommons.org/licenses/by-nc-nd/4.0/deed.en)

2023-03-09

Formal Concept Analysis

Formal Concept Analysis
The Galactic Organization <contact@thegalactic.org>
2018-2023
© 2018-2023 the Galactic Organization. This document is licensed under CC-by-nc-nd (https://creativecommons.org/licenses/by-nc-nd/4.0/deed.en)

Formal Concept Analysis

Basic elements

- concepts
- context

Formal Concept Analysis
Basic elements
▶ concepts
▶ context

- Formal Concept Analysis (FCA) involves the definition of concepts in a given context. These concepts and contexts are completely and precisely defined.
- FCA was introduced in 1982 by Rudolf Wille [^ganter1999afc] as an application of lattice theory which is based on the works of M. Barbut and B. Monjardet on the whole lattice theory and on Galois connections [^barbut1970].

Formal Concept Analysis

Basic elements

- concepts
- context

History

- introduced in 1982 by Rudolf Wille
- application of **lattice theory** which is based on the works of M. Barbut and B. Monjardet on the whole lattice theory and on Galois connections.

Formal Concept Analysis
Basic elements
concepts
context
History
introduced in 1982 by Rudolf Wille
application of **lattice theory** which is based on the works of M. Barbut and B. Monjardet on the whole lattice theory and on Galois connections.

- Formal Concept Analysis (FCA) involves the definition of concepts in a given context. These concepts and contexts are completely and precisely defined.
- FCA was introduced in 1982 by Rudolf Wille [^ganter1999afc] as an application of lattice theory which is based on the works of M. Barbut and B. Monjardet on the whole lattice theory and on Galois connections [^barbut1970].

Comparison with (deep) learning methods

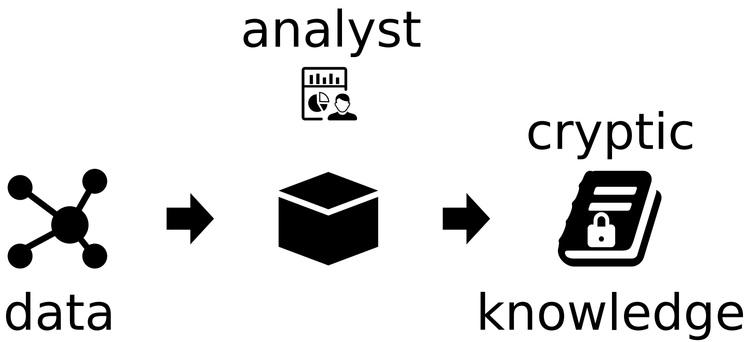
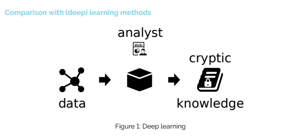


Figure 1: Deep learning



Comparison with (deep) learning methods

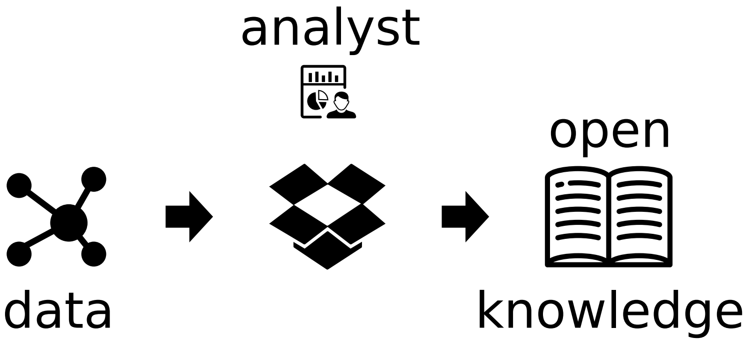
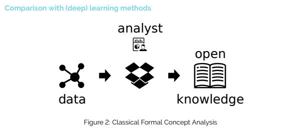


Figure 2: Classical Formal Concept Analysis

2023-03-09



Comparison with (deep) learning methods

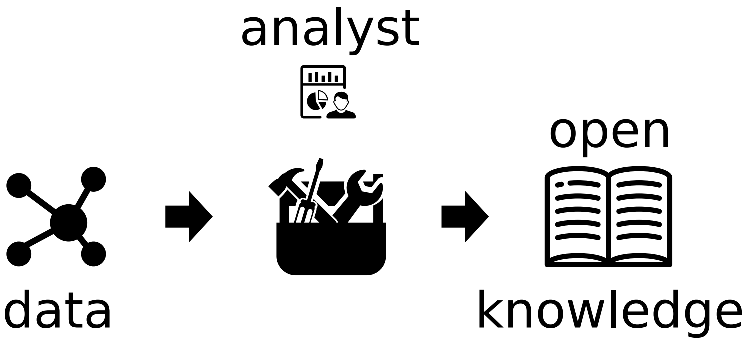
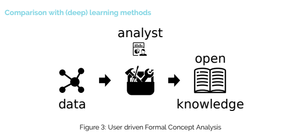


Figure 3: User driven Formal Concept Analysis

2023-03-09



Applications

FCA allows to analyze and extract information from a context where implicit knowledge regarding a dataset can be represented.

- FCA is used in many fields, such as
  - data mining;
  - machine learning;
  - information retrieval;
  - knowledge management.
- with applications in:
  - text mining;
  - semantic web;
  - software development;
  - chemistry and biology.

Applications
FCA allows to analyze and extract information from a context where implicit knowledge regarding a dataset can be represented.
FCA is used in many fields, such as data mining; machine learning; information retrieval; knowledge management.
with applications in: text mining; semantic web; software development; chemistry and biology.

it is used in many fields!

Applications

- it is used for **classification** and **clustering** because it allows the association of several objects to the same description;
- it is also used to extract **association rules** that are important in the field of data mining, and which are used by many applications.

Applications

- it is used for **classification** and **clustering** because it allows the association of several objects to the same description;
- it is also used to extract **association rules** that are important in the field of data mining, and which are used by many applications.

Example

A context is represented by a two-dimensional table, which expresses the binary relationship between objects and their attributes.

Here we have animals and their attributes expressed by a context.

Animals	small	medium	big	twolegs	fourlegs	feathers
Dove	✓			✓		✓
Hen	✓			✓		✓
Tiger			✓		✓	
Lion			✓		✓	
Fox		✓			✓	
Dog		✓			✓	

To understand, let's see this example;

Example

- a lattice is represented with a graph;
- every pair of nodes have a:
  - **common upper bound**;
  - **common lower bound**.
- nodes are concepts;
- edges express the **generalization/specialization** relation.

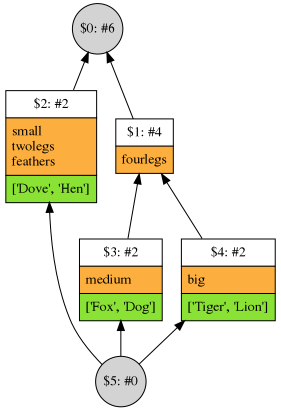
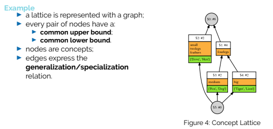


Figure 4: Concept Lattice



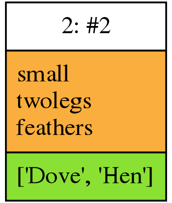
The picture below represents the lattice representation of the context.

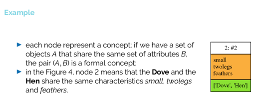
- every pair of nodes have a common upper bound. defining the shared attributes;
- and a common lower bound, defining the shared objects.

It is also important to specify here that the edges express the generalization/specialization relationship.
node 1 is a generalization of nodes 3 and 4, so 3 and 4 contains both the attribute "fourlegs".

Example

- each node represent a concept; if we have a set of objects A that share the same set of attributes B, the pair (A, B) is a formal concept;
- in the Figure 4, node 2 means that the **Dove** and the **Hen** share the same characteristics *small*, *twolegs* and *feathers*.





so you can see here that the lattice is some kind of hierarchical representation of the data,

So, let's move to formal definitions!

Context, Concept and Lattice

Definition

A formal context is a triple $\langle G, M, I \rangle$, G is a set of object (*Gegenstand*), M is a set of attributes (*Merkmal*), and I is a relation between G and M (*Incidence*); if an object $g \in G$ has a relation with an attribute $m \in M$ we write gIm and we say that the object g has the attribute m.

Context, Concept and Lattice
Definition
A formal context is a triple $\langle G, M, I \rangle$, G is a set of object (*Gegenstand*), M is a set of attributes (*Merkmal*), and I is a relation between G and M (*Incidence*); if an object $g \in G$ has a relation with an attribute $m \in M$ we write gIm and we say that the object g has the attribute m.

Galois Connection

A Galois connection $\langle \alpha, \beta \rangle$ is defined as follows:

Definition

$$\alpha: \quad 2^G \to 2^M$$
$$G \supseteq X \mapsto B \subseteq M$$

associates common attributes B to a subset of objects X

Definition

$$\beta: \quad 2^M \to 2^G$$
$$M \supseteq Y \mapsto A \subseteq G$$

associates shared objects A to a subset of attributes Y

2023-03-09
Formal Concept Analysis
└Formal Concept Analysis
└Context, Concept and Lattice
└Galois Connection

Galois Connection
A Galois connection ⟨α, β⟩ is defined as follows:
Definition
α: 2^G → 2^M
G ⊇ X ↦ B ⊆ M
associates common attributes B to a subset of objects X
Definition
β: 2^M → 2^G
M ⊇ Y ↦ A ⊆ G
associates shared objects A to a subset of attributes Y

Galois Connection

Examples

$$\alpha(\{\text{Fox}\}) = \{\text{medium}, \text{fourlegs}\}$$

$$\alpha(\{\text{Fox}, \text{Dog}\}) = \{\text{medium}, \text{fourlegs}\}$$

$$\beta(\{\text{twolegs}\}) = \{\text{Dove}, \text{Hen}\}$$

$$\beta(\emptyset) = \{\text{Dove}, \text{Hen}, \text{Tiger}, \text{Lion}, \text{Fox}, \text{Lion}\}$$

Galois Connection
Examples
α({Fox}) = {medium, fourlegs}
α({Fox, Dog}) = {medium, fourlegs}
β({twolegs}) = {Dove, Hen}
β(∅) = {Dove, Hen, Tiger, Lion, Fox, Lion}

Closure operators

Definition

$$\alpha \circ \beta : \begin{array}{ll} 2^M & \to 2^M \\ M \supseteq Y & \mapsto B \subseteq M \end{array} \qquad \beta \circ \alpha : \begin{array}{ll} 2^G & \to 2^G \\ G \supseteq X & \mapsto A \subseteq G \end{array}$$

are closure operators

2023-03-09

Closure operators
Definition
are closure operators

Closure operators

Examples

$$\beta \circ \alpha(\{\text{Fox}\}) = \{\text{Fox}, \text{Dog}\}$$

$$\beta \circ \alpha(\{\text{Fox}, \text{Dog}\}) = \{\text{Fox}, \text{Dog}\}$$

$$\alpha \circ \beta(\{\text{twolegs}\}) = \{\text{small}, \text{twolegs}, \text{feathers}\}$$

$$\alpha \circ \beta(\{\text{small}, \text{twolegs}, \text{feathers}\}) = \{\text{small}, \text{twolegs}, \text{feathers}\}$$

2023-03-09

Closure operators
Examples
$\beta \circ \alpha(\{\text{Fox}\}) = \{\text{Fox}, \text{Dog}\}$
$\beta \circ \alpha(\{\text{Fox}, \text{Dog}\}) = \{\text{Fox}, \text{Dog}\}$
$\alpha \circ \beta(\{\text{twolegs}\}) = \{\text{small}, \text{twolegs}, \text{feathers}\}$
$\alpha \circ \beta(\{\text{small}, \text{twolegs}, \text{feathers}\}) = \{\text{small}, \text{twolegs}, \text{feathers}\}$

Concepts

Definition

A concept is a pair ($A \subseteq G, B \subseteq M$)

$$B = \alpha(A) \ \wedge \ A = \beta(B)$$

$\leq$: binary relation of "specialisation/generalisation" between concepts

$$(A_1, B_1) \leq (A_2, B_2) \iff A_1 \subseteq A_2 \iff B_2 \subseteq B_1$$

Concepts
Definition
A concept is a pair ($A \subseteq G, B \subseteq M$)
$B = \alpha(A) \wedge A = \beta(B)$
$\leq$: binary relation of "specialisation/generalisation" between concepts
$(A_1, B_1) \leq (A_2, B_2) \iff A_1 \subseteq A_2 \iff B_2 \subseteq B_1$

alpha, is an operator from a set of objects to a set of attributes
(α, β) is a Galois connection between 2^G and 2^M if:

- α is an antitone map from G to M
- β is an antitone map from M to G
- $(\beta \ o \ \alpha)$ and $(\alpha \ o \ \beta)$ are closure operators.

A concept is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $B = \alpha(A)$ *and* $A = \beta(B)$
$\leq$: binary relation of "specialisation/generalisation" between concepts

$$(A1, B1) \leq (A2, B2) \Leftrightarrow B1 \in B2 \text{ and } A1 \in A2$$

FCA, Tools and Algorithms

In **FCA**, we talk generaly about three structures, or data forms:

- **Reduced Context** (or data): is a two dimension table showing the relation between objects and attributes;
- **Lattice**: is a graph, where every node is a concept represent a set of object sharing the same set of attributes;
- **Basis of rules**: are implications, showing the relation between attributes.

These three structures represent the same information, but in different ways.

FCA, Tools and Algorithms
In FCA, we talk generaly about three structures, or data forms:
Reduced Context (or data): is a two dimension table showing the relation between objects and attributes;
Lattice: is a graph, where every node is a concept represent a set of object sharing the same set of attributes;
Basis of rules: are implications, showing the relation between attributes.
These three structures represent the same information, but in different ways.

- Basis of rules: minimal set of implications rules between attributes

FCA, Tools and Algorithms

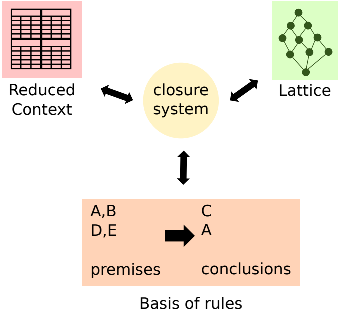
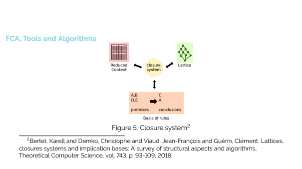


Figure 5: Closure system[2]

[2] Bertet, Karell and Demko, Christophe and Viaud, Jean-François and Guérin, Clément. Lattices, closures systems and implication bases: A survey of structural aspects and algorithms, Theoretical Computer Science, vol. 743, p. 93-109, 2018.



- Many algorithms allow transformation and generate one representation from another,

Pattern theory

- pattern structures allows to extend **FCA** algorithms to non-binary data when a *Galois connection* exists between objects and their descriptions;
- *the idea behind the use of pattern is that we don't have always disassociated attributes; patterns are some kind of descriptions that the objects have, and of course we need to have some kind of similarity function between descriptions telling at which point two objects are similar.* [3]

[3]GANTER, Bernhard et KUZNETSOV, Sergei O. Pattern structures and their projections. In: International Conference on Conceptual Structures. Springer, Berlin, Heidelberg, 2001. p. 129-142.

Pattern theory

- pattern structures allows to extend **FCA** algorithms to non-binary data when a *Galois connection* exists between objects and their descriptions;
- *the idea behind the use of pattern is that we don't have always disassociated attributes; patterns are some kind of descriptions that the objects have, and of course we need to have some kind of similarity function between descriptions telling at which point two objects are similar.* [3]

[3]GANTER, Bernhard et KUZNETSOV, Sergei O. Pattern structures and their projections. In: International Conference on Conceptual Structures. Springer, Berlin, Heidelberg, 2001. p. 129-142.