Formal Concept Analysis in a Nutshell
<table>
<thead>
<tr>
<th>Outline</th>
<th>Concept lattices</th>
<th>Attribute logic</th>
<th>Many valued contexts</th>
<th>Conclusion</th>
</tr>
</thead>
</table>

### 1 Concept lattices
- Data from a hospital
- Formal definitions
- More examples
Outline

1. Concept lattices
   - Data from a hospital
   - Formal definitions
   - More examples

2. Attribute logic
   - Checking completeness
Outline

1. Concept lattices
   - Data from a hospital
   - Formal definitions
   - More examples

2. Attribute logic
   - Checking completeness

3. Many valued contexts
   - Scaling
   - Turtoise logic

Formal Concept Analysis in a Nutshell
Outline

1. Concept lattices
   - Data from a hospital
   - Formal definitions
   - More examples

2. Attribute logic
   - Checking completeness

3. Many valued contexts
   - Scaling
   - Turtoise logic

4. Conclusion
## Data from a hospital

### Interview data from a treatment of Anorexia nervosa

<table>
<thead>
<tr>
<th></th>
<th>over-sensitive</th>
<th>withdrawn</th>
<th>self-confident</th>
<th>dutiful</th>
<th>cordial</th>
<th>difficult</th>
<th>attentive</th>
<th>easily offended</th>
<th>calm</th>
<th>apprehensive</th>
<th>chatty</th>
<th>superficial</th>
<th>sensitive</th>
<th>ambitious</th>
</tr>
</thead>
<tbody>
<tr>
<td>Myself</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>My Ideal</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Father</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Mother</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Sister</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Brother-in-law</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td></td>
<td>✗</td>
<td></td>
<td>✗</td>
</tr>
</tbody>
</table>
A biplot of the interview data

- Brother-in-law
- My Ideal
- Myself
- Mother
- Father
- Sister

Attributes:
- uncomplicated
- valiant
- thick-skinned
- sympathetic
- attentive
- superficial
- complicated
- difficult
- apprehensive
- over-sensitive
Data from a hospital

The concept lattice of the interview data
Unfolding data in a concept lattice

The basic procedure of Formal Concept Analysis:

- Data is represented in a very basic data type, called **formal context**.
Unfolding data in a concept lattice

The basic procedure of Formal Concept Analysis:

- Data is represented in a very basic data type, called **formal context**.
- Each formal context is transformed into a mathematical structure called **concept lattice**.
Formal definitions

Unfolding data in a concept lattice

The basic procedure of Formal Concept Analysis:

- Data is represented in a very basic data type, called **formal context**.

- Each formal context is transformed into a mathematical structure called **concept lattice**. The information contained in the formal context is preserved.
Unfolding data in a concept lattice

The basic procedure of Formal Concept Analysis:

- Data is represented in a very basic data type, called **formal context**.
- Each formal context is transformed into a mathematical structure called **concept lattice**. The information contained in the formal context is preserved.
- The concept lattice is the basis for further data analysis. It may be represented graphically to support communication, or it may be investigated with algebraic methods to unravel its structure.
The divisor lattice of 200
Recommended serving temperatures of some red wines

- **15° C**: Trollinger, Beaujolais, Burgundy
- **16° C**: Bordeaux
- **17° C**: Barbera
- **18° C**: Negroamaro
- **19° C**: Brunello
Recommended serving temperatures for white wines

- 6°C: Moscato, Cava, Prosecco
- 7°C: Moscato, Cava, Prosecco
- 8°C: Moscato, Cava, Prosecco, Pinot grigio, Vernaccia
- 9°C: Moscato, Cava, Prosecco, Pinot grigio, Vernaccia, Sauvignon blanc, Riesling, Chardonnay, white Burgundy
- 10°C: Moscato, Cava, Prosecco, Pinot grigio, Vernaccia, Sauvignon blanc, Riesling, Chardonnay, white Burgundy
- 11°C: Moscato, Cava, Prosecco, Pinot grigio, Vernaccia, Sauvignon blanc, Riesling, Chardonnay, white Burgundy, white Rhone
- 12°C: Moscato, Cava, Prosecco, Pinot grigio, Vernaccia, Sauvignon blanc, Riesling, Chardonnay, white Burgundy, white Rhone, Gewürztraminer
- 13°C: Moscato, Cava, Prosecco, Pinot grigio, Vernaccia, Sauvignon blanc, Riesling, Chardonnay, white Burgundy, white Rhone, Gewürztraminer
- 14°C: Moscato, Cava, Prosecco, Pinot grigio, Vernaccia, Sauvignon blanc, Riesling, Chardonnay, white Burgundy, white Rhone, Gewürztraminer
## An example about airlines...

<table>
<thead>
<tr>
<th>Country/Company</th>
<th>Latin America</th>
<th>Europe</th>
<th>Canada</th>
<th>Asia Pacific</th>
<th>Middle East</th>
<th>Africa</th>
<th>Mexico</th>
<th>Caribbean</th>
<th>United States</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air Canada</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Air New Zealand</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All Nippon Airways</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ansett Australia</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The Austrian Airlines Group</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>British Midland</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lufthansa</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mexicana</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scandinavian Airlines</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Singapore Airlines</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thai Airways International</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>United Airlines</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VARIG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
... and its concept lattice
Formal Concept Analysis?

- Branch of Applied Mathematics
- Based on *Lattice Theory* developed by Garrett Birkhoff and others in the 1930s
- Employs algebra in order to formalize notions of *concept* and *conceptual hierarchy*
- Term *Formal Concept Analysis* (short: FCA) introduced by Rudolf Wille in the 1980s.
Why Formal Concept Analysis?

- The method of Formal Concept Analysis offers an algebraic approach to **data analysis** and **knowledge processing**.

- Strengths of FCA are
  - ... a solid mathematical and philosophical foundation,
  - ... more than 1000 research publications,
  - ... experience of several hundred application projects,
  - ... an expressive and intuitive graphical representation,
  - and a good algorithmic basis.

- Due to its elementary yet powerful formal theory, FCA can express other methods, and therefore has the potential to unify the methodology of **data analysis**.
Introduction

Formal Concept Analysis (FCA) is a...

- “mathematization” of the philosophical understanding of concepts
- Human-centered method to structure and analyze data
- Method to visualize data and its inherent structures, implications and dependencies

The following slides are partially based on the ICFCA’04 tutorial by Bastian Wormuth and Peter Becker http://www.wormuth.info/ICFCA04/materials.html
What is a Concept?

- Consider the concept “bird”. What drives us to call something a “bird”?
- Every object with certain attributes is called “bird”:
  - A bird has **feathers**.
  - A bird has **two legs**.
  - A bird has a **bill**. …
- All **objects** having these attributes are called “birds”:
  - Duck, goose, owl and parrot are birds.
  - Penguins are birds, too.
  - …
What is a Concept?

- This description of the concept “bird” is based on sets of objects related to attributes.

**Objects**
- duck
- goose
- parrot
- ...

**Attributes**
- has bill
- has feathers
- has two legs
- ...

Objects, attributes and a relation form a **formal concept**.
What is a Concept?

- So, a **formal concept** is constituted by two parts
  
  - **A**: a set of objects
  - **B**: a set of attributes

- ... having a certain relation:
  
  - every **object** belonging to this concept has **all** the attributes in **B**
  - every **attribute** belonging to this concept is shared by **all** objects in **A**

- **A** is called the concept's **extent**, **B** is called the concept's **intent**
A repertoire of **objects** and **attributes** (which might or might not be related) constitutes the „context“ of our considerations.
## The Formal Context

<table>
<thead>
<tr>
<th>K</th>
<th>small</th>
<th>medium</th>
<th>big</th>
<th>2legs</th>
<th>4legs</th>
<th>feathers</th>
<th>hair</th>
<th>fly</th>
<th>hunt</th>
<th>run</th>
<th>swim</th>
<th>mane</th>
<th>hooves</th>
</tr>
</thead>
<tbody>
<tr>
<td>dove</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hen</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>duck</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>goose</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>owl</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hawk</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eagle</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fox</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dog</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wolf</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cat</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tiger</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lion</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>horse</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>zebra</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cow</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**set of objects** (G)

**set of attributes** (M)

Crosses indicate incidence relation (I) between G and M

(G,M,I) is called **formal context**
Definition of Formal Concepts

For the mathematical definition of formal concepts we introduce the derivation operator “‘”.

For a set of objects A, A’ is defined as:
\[ A' = \{\text{all attributes in } M \text{ common to the objects of } A\} \]

For a set of attributes B, B’ is defined as:
\[ B' = \{\text{objects in } G \text{ having all attributes of } B\} \]

We are looking for pairs (A,B) of objects A and attributes B that satisfy the conditions
\[ A' = B \text{ and } B' = A \]
and we call these pairs formal concepts.
Calculating Formal Concepts

- Using the derivation operator we can derive formal concepts from our formal context with the following procedure:
  - Pick an object set $A$.
  - Derive the attributes $A'$.
  - Derive $(A')'$.
  - $(A'',A')$ is a formal concept.

- The same routine could be applied starting with an attribute set $B$: $(B',B'')$ is a formal concept as well.
Calculating Formal Concepts

1. Pick a set of objects: \( A = \{\text{duck}\} \)
2. Derive attributes: \( A' = \{\text{small, 2legs, feathers, fly, swim}\} \)
3. Derive objects: \( (A')' = \{\text{small, 2legs, feathers, fly, swim}\}' = \{\text{duck, goose}\} \)
4. Formal concept: \( (A'',A') = (\{\text{duck, goose}\}, \{\text{small, 2legs, feathers, fly, swim}\}) \)
Ordering Concepts

The formal concept \((A'',A')=(\{\text{duck, goose}\}, \{\text{small, 2legs, feathers, fly, swim}\})\) is represented in the line diagram as a node:

\[B' = \{\text{duck, goose, dove, owl, hawk}\}\]
\[B'' = \{\text{small, 2legs, feathers, fly}\}\]

\[A'' = \{\text{duck, goose}\}\]
\[A' = \{\text{small, 2legs, feathers, fly, swim}\}\]

Consider another formal concept \((B',B'')=(\{\text{duck, goose, dove, owl, hawk}\}, \{\text{small, 2legs, feathers, fly}\})\).

The formal concept \((B',B'')\) is a superconcept of \((A'',A')\) and \((A'',A')\) is a subconcept of \((B',B'')\), because \(A''\) is a subset of \(B'\).

So \((B',B'')\) is drawn above \((A'',A')\) and connected to it by a line.
Ordering Concepts

We extend the diagram by adding more formal concepts
({owl, hawk}, {feathers, 2legs, small, fly, hunt})
({owl, hawk, eagle}, {feathers, 2legs, fly, hunt})

... and subconcept relations:

... and so on.

Several methods exist to derive all formal concepts:
Cut over extents, Ganter's algorithm etc.
The subconcept–superconcept relation defines an order \( \leq \) on the set \( B \) of all formal concepts of a formal context. For two concepts \((A_1, A_2)\) and \((B_1, B_2)\), this order is defined by:

\[
(A_1, A_2) \leq (B_1, B_2) \iff A_1 \subseteq B_1 (\iff B_2 \subseteq A_2)
\]

\((A_1, A_2)\) is smaller than \((B_1, B_2)\) if \(A_1\) is subset of \(B_1\) (objects) and \(B_2\) is subset of \(A_2\) (attributes). Hence, \((B, \leq)\) is an ordered set.

The set \( B \) of formal concepts has another property:

- For each family of formal concepts of a formal context there exists always a unique greatest subconcept and a unique smallest superconcept.
- The ordered set \( B = (B, \leq)\) plus the last property forms a mathematical structure: the concept lattice.
Concept Lattice – Formal Concepts

{duck, goose, dove, owl, hawk, eagle}
{2legs, feathers, fly}

{duck, goose, dove, owl, hawk}
{small, 2legs, feathers, fly}

„birds“
„medium hunting birds“

{eagle}
{medium, hunt, 2legs, feathers, fly}

{duck, goose}
{small, swim, 2legs, feathers, fly}
Concept Lattice – Top and Bottom

- More general concepts lead to more specific concepts.
- Attributes and objects are distributed throughout the lattice.
- The top of the lattice has no attributes and all objects.
- The bottom of the lattice has all attributes and no objects.
More examples

Formal concepts

\[(A, B)\] is a **formal concept** of \((G, M, I)\) iff

\[A \subseteq G, \quad B \subseteq M, \quad A' = B, \quad A = B'.\]

\(A\) is the **extent** and \(B\) is the **intent** of \((A, B)\).
Formal concepts can be ordered by

\[(A_1, B_1) \leq (A_2, B_2) : \iff A_1 \subseteq A_2.\]

The set \( \mathcal{B}(G, M, I) \) of all formal concepts of \((G, M, I)\), with this order, is a complete lattice, called the **concept lattice** of 
\((G, M, I)\).
More examples

### A formal context

<table>
<thead>
<tr>
<th></th>
<th>needs water to live</th>
<th>lives in water</th>
<th>lives on land</th>
<th>needs chlorophyll</th>
<th>two seed leaves</th>
<th>one seed leaf</th>
<th>can move around</th>
<th>has limbs</th>
<th>suckles its offspring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leech</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bream</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>Frog</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>Dog</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Spike-weed</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td>×</td>
<td></td>
<td>×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reed</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bean</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maize</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
More examples

and its concept lattice

Formal Concept Analysis in a Nutshell
Theorem 3 (The Basic Theorem on Concept Lattices). The concept lattice $\mathfrak{B}(G, M, I)$ is a complete lattice in which infimum and supremum are given by:

$$\bigwedge_{t \in T} (A_t, B_t) = \left( \bigcap_{t \in T} A_t, \left( \bigcup_{t \in T} B_t \right)'' \right)$$

$$\bigvee_{t \in T} (A_t, B_t) = \left( \left( \bigcup_{t \in T} A_t \right)'' , \bigcap_{t \in T} B_t \right).$$

A complete lattice $V$ is isomorphic to $\mathfrak{B}(G, M, I)$ if and only if there are mappings $\tilde{\gamma} : G \rightarrow V$ and $\tilde{\mu} : M \rightarrow V$ such that $\tilde{\gamma}(G)$ is supremum-dense in $V$, $\tilde{\mu}(M)$ is infimum-dense in $V$ and $gI m$ is equivalent to $\tilde{\gamma}g \leq \tilde{\mu}m$ for all $g \in G$ and all $m \in M$. In particular, $V \cong \mathfrak{B}(V, V, \leq)$. 
Applications?

Formal Concept Analysis has recently been applied in

- Description Logics, for checking completeness of knowledge bases,
- Linguistics, for the investigation of thesauri and ontologies,
- Software Engineering, for modelling type hierarchies with role types,
- Biomathematics, for analysing gene expression data,
- Machine Learning, for discovering website duplicates,
- Data Mining, for pattern matching problems,
- Rough Set Theory, for studying granular data,
- et cetera . . .