GALACTIC

K. Bertet and C. Demko

Objective: The objective of this practical work is to generate explainable concepts/clusters using GALACTIC and to visualize them with Tableau

1. Installation and get started with GALACTIC

Exercice 1. The first step will be to install GALACTIC, available at galactic.univ-lr.fr, with in particular:

- An installation guide in https://galactic.univ-lr.fr/docs/archives/main/latest/install/
- A complete documentation of the tool in galactic.univ-lr.fr/docs
- User guides, with in particular a collection of jupyter notebooks of the core and the different plugins in galactic.univ-lr.fr/guides

The installation can be done either within a virtual environment or in the ~/.local/ directory.

- For the virtual environment, it is necessary to create it using, if necessary, the system of python packages:

```
$ python -m venv galactic --system-site-packages
$ source galactic/bin/activate
(galactic) $ pip install --find-links https://galactic.univ-lr.fr/packages
py-galactic[docs]
```

- For the local installation in the ~/.local/ directory :

```
$ pip install --find-links https://galactic.univ-lr.fr/packages py-galactic
[docs]
```

Use the [.docs] option during installation, then check that GALACTIC is correctly installed.

Analysis of the dataset digits with python

Exercice 2. Run the following python code which loads the digits dataset, generates the concept lattice by specifying the descriptions and strategies to use, then displays it:

```
from galactic_description_logical_classic import BooleanDescription
from galactic_characteristic_logical import Bool
from galactic.algebras.lattice import *
from galactic.algebras.poset import HasseDiagram
from galactic.population import Population
from galactic.descriptions import Member
from galactic.concepts import ConceptLattice, Concept, ConceptRenderer
from galactic.strategies import *
from galactic.characteristics import *
from galactic.examples import *
from galactic.concepts import ConceptLattice
from galactic_strategy_logical_classic_basic import BooleanStrategy

# the data
data = {
0: [“c”, “e”, “s”],
1: ["o", "s"],
```

```
2: ["e", "p"],
3: ["o", "p"],
4: ["c", "e", "s"],
5: ["o", "p"],
6: ["c", "e"],
7: ["o", "p"],
8: ["c", "e"],
9: ["c", "o", "s"],
}
## creating a population from the data
population = Population(data)
## creating descriptions
descriptions = [
        BooleanDescription(Bool(Member(name="c"))),
        BooleanDescription(Bool(Member(name="e"))),
        BooleanDescription(Bool(Member(name="o"))),
        BooleanDescription(Bool(Member(name="p"))),
        BooleanDescription(Bool(Member(name="s"))),
]
## creating a concept lattice based on the population and the descriptions
lattice = ConceptLattice(population, descriptions)
## using the stragtegies to generate the lattice
lattice.apply([
        BooleanStrategy(Bool(Member(name="c"))),
        BooleanStrategy(Bool(Member(name="e"))),
        BooleanStrategy(Bool(Member(name="o"))),
        BooleanStrategy(Bool(Member(name="p"))),
        BooleanStrategy(Bool(Member(name="s"))),
])
## showing the hasse diagram
HasseDiagram(lattice, domain_renderer=ConceptRenderer())
```

Exercice 3. Identify the methods allowing to display the number of concepts of the lattice. How many concepts are generated? Print some concepts/subgroups that seem relevant to you and identify the objects that compose them as well as the predicates describing them.

Exercice 4. The lattice can be represented by a binary table where the objects (in row) are described by the predicates (in columns) computed by the descriptions. Run the following code and identify the set of all generated predicates.

```
BinaryTable(
    lattice.reduced_context,
    domain_renderer=ConceptRenderer(join_irreducible=True),
    co_domain_renderer=ConceptRenderer(meet_irreducible=True)
)
```

Analysis of the dataset digits with Galactic-laser

Exercice 5. Create the *digits* dataset in csv format, then load it with the *galactic-laser-iu* tool. Describe the desired analysis with the following yaml exploration file which specifies that each characteristic is a boolean characteristic of plugin *characteristic.logical.Bool*, the object descriptions will be classic set descriptions of plugin description.logical.classic.Boolean, and exploration strategies will be classic boolean strategies of plugin strategy.logical.classic.basic.Boolean. Generate the subgroups

```
characteristics:
  - &id001 !characteristic.logical.Bool
    - !predicate.core.Member
      name: c
  - &id002 !characteristic.logical.Bool
    - !predicate.core.Member
      name: e
  - &id003 !characteristic.logical.Bool
    - !predicate.core.Member
      name: o
  - &id004 !characteristic.logical.Bool
    - !predicate.core.Member
      name: p
  - &id005 !characteristic.logical.Bool
    - !predicate.core.Member
      name: s
descriptions:
  - !description.logical.classic.Boolean
    - *id001
  - !description.logical.classic.Boolean
    - *id002
  - !description.logical.classic.Boolean
    - *id003
  - !description.logical.classic.Boolean
    - *id004
  - !description.logical.classic.Boolean
    - *id005
strategies:
  - !strategy.logical.classic.basic.Boolean
    - *id001
  - !strategy.logical.classic.basic.Boolean
    - *id002
  - !strategy.logical.classic.basic.Boolean
    - *id003
  - !strategy.logical.classic.basic.Boolean
    - *id004
  - !strategy.logical.classic.basic.Boolean
    - *id005
```


Analysis of the supervised dataset Lenses of categorical data

Consider the Lenses dataset issued from the UCI Machine Learning Repository and available here : *~/.local/share/galactic/sample/data/lenses/*lenses.csv .

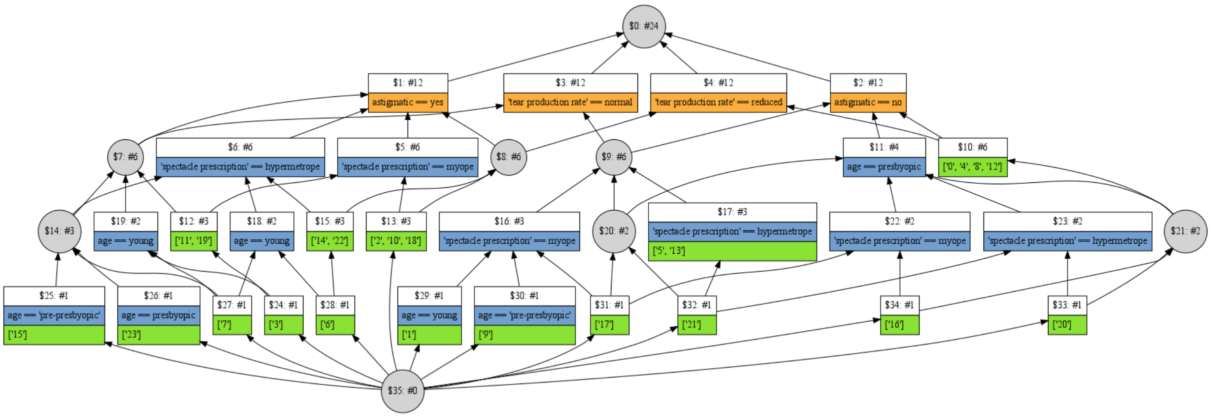
This dataset is composed of 24 objects/patients described by 4 categorical attributes :

- age of the patient (G) : young (y) ; pre-presbyopic (pp) ; presbyopic (p)
- prescription (P) : myope (m) ; hypermetrope (h)
- astigmatic (A) : no (n) ; yes (y)
- tear production rate (T): reduced (r) normal (n)

and classified in 3 classes (C) :

1. the patient should be fitted with hard contact lenses (h)
2. the patient should be fitted with soft contact lenses (s)
3. the patient should not be fitted with contact lenses (n).

Exercice 6. Load these data with Galactic, then analyze them using the following exploration file which specify a classical binary description by a set of attributes, and the supervised entropy strategy :
~/.local/share/galactic/sample/data/lenses/explorer-entropy.yaml
You should get this lattice :



Exercice 7. Identify the concepts containing objects that all belong to the same class. These concepts represent clusters of the data. Extract their description.

Exercice 8. <u>Go back to Tableau:</u> Add a new attribute *cluster* in the initial dataset *Lenses* which specify the relevant clusters composed of objects belonging to the same class. Then propose a visualization of the dataset using this new attribute.

Exercice 9. [To go further] Modify the analysis to use the logical description of binary data. What are the descriptions generated? Identify the generated clusters that are composed of patients of the same class, as well as their description.

Analysis of the supervised dataset Iris of numerical data

The iris dataset Iris dataset is issued from the UCI Machine Learning Repository , and is composed of 150 objects described by 4 numerical characteristics:

- sepal-length, sepal-width, petal-length, petal-width

and classified in 3 classes:

- Setosa, Versicolor, Virginica

Exercice 10. Generate with Galactic, and more precisely with an exploration based of the entropy meta-strategy, the concept lattice in Figure 1.

Exercice 11. Identify the 5 concepts composed of objects belonging to the same class and print their description

Exercice 12. <u>Go back to Tableau:</u> Add a new attribute *cluster* in the initial dataset *Iris* which specify the concept/cluster composed of objects belonging to the same class, and other new attributes which specify its description. Then propose a visualization of the dataset using these new attributes.

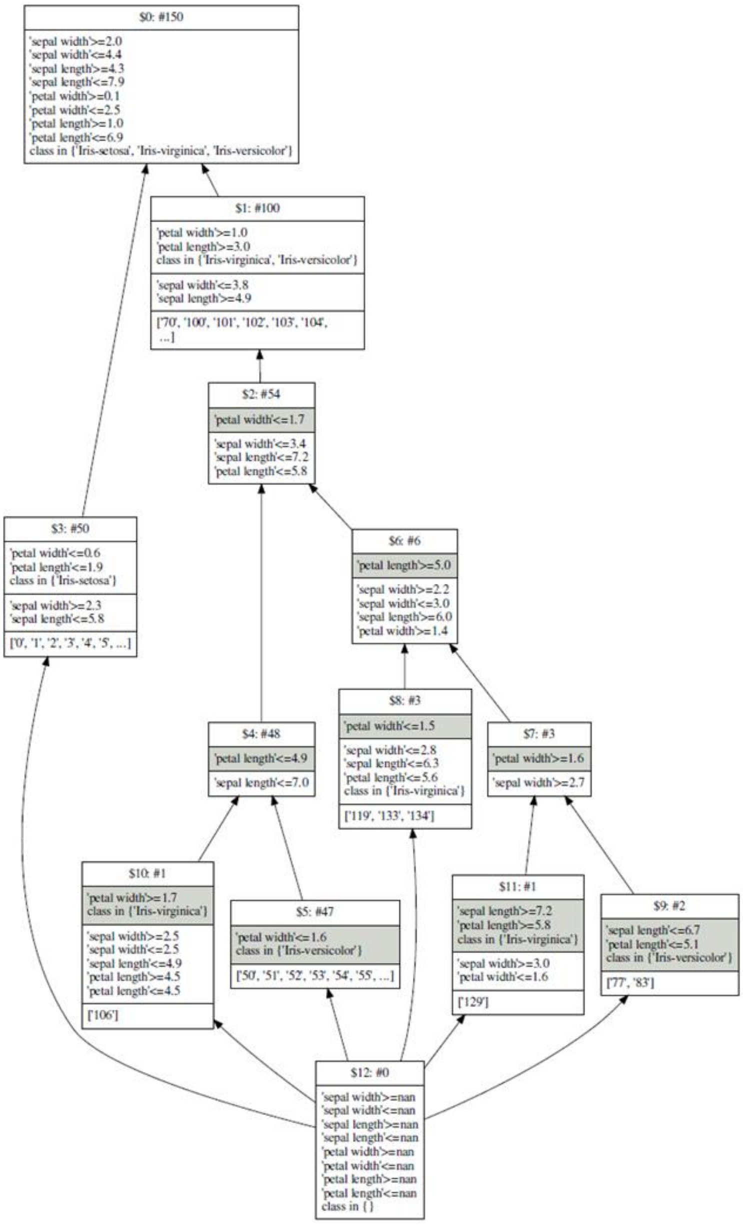


Figure 1 Concept lattice for the iris dataset