IA301 - Practical work on decision trees and ontologies

October 2022

This work has two parts:

1. decision trees - report to be delivered as a Jupyter Notebook with detailed comments and answers to the questions;
2. ontology - report to be delivered as a pdf file, including snapshots of the ontology.

Both reports should be put on the course moodle. They can be written either in French or in English. This is a group work, including for the reports, and groups can include 4-6 students.

1 Decision trees

This part aims at experimenting with the decision tree method for classification available in ScikitLearn (`https://scikit-learn.org/stable/modules/tree.html`) on simple examples. First look at the documentation to understand the main function and its parameters.

Use the proposed Jupyter Notebook (e.g. in the Anaconda distribution, python 3): `DecisionTree.ipynb`, run the examples and change the code to answer the questions. Questions can be answered directly in the notebook, either in French or in English.

2 Ontologies

First download the desktop version of protege from `https://protege.stanford.edu/`, and launch it.

You can read `http://protegeproject.github.io/protege/getting-started/`

2.1 Simple ontology

This part aims at understanding a few functionalities of protege.

- Class hierarchy: use "Entities / Classes" to create
  - Person as a subclass of owl:Thing
  - Man, Woman, and Parent as subclasses of Person
  - Father as a subclass of Man and Parent (use "Subclass of" in the right panel)
  - Add the constraint that Man and Woman are disjoint (use "Disjoint with" in the right panel)
- Class properties: use "Data properties" to create
  - age with domain Person and range xsd:int

  use "Object properties" to create

  - isFriendOf with domain Person and range Person, and that is symmetric (in “Characteristics” panel)
  - isParentOf with domain Person and range Person
  - isFatherOf as subproperty of isParentOf with domain Man and range Person
  - is ChildOf with domain Person and range Person, and is the inverse property of isParentOf

- Individuals: use “Individuals” to create
  - John, in the Man class, 30 years old, isFriendOf Lea
  - Lea, 31 years old
  - Tom, 5 years old, isChildOf John and Lea

- Start reasoner (e.g. HermiT): this will check the consistency of the ontology.

- Switch to Inferred in class hierarchy and check the classes. What do you observe?

- Use “DL queries” to execture a few queries (check the appropriate boxes on the right), such as:
  - Person and isFriendOf some
  - Person and isParentOf some
  - age value 30
  - age some xsd:integer[>20]
  - ...

  (syntax and examples can be found on `https://protegewiki.stanford.edu/wiki/DLQueryTab`)

2.2 Create your own ontology

Choose a problem (for instance for sustainability, such as water consumption)[1]. Explain the design of the ontology (classes, data properties, object properties, class axioms, individuals...). Provide snapshots of the ontolgy (asserted and then inferred), and explain some queries[2].

[1] You can get inspiration from `https://protegewiki.stanford.edu/wiki/Protege4Pizzas10Minutes`

[2] You may find useful information on the FAQ collected by Natalia Diaz during the last years: `https://docs.google.com/document/d/1rpPh69auU4PJ9Wh0b6QMp9IlhHflkN7EjLrq353OSbA/edit`