Assignment 4: Reasoning & Decision Trees

Artificial Intelligence

WS 2023

Due: 2023-12-11, 12:00 (noon)

Katharina Hoedt
Verena Praher
Florian Schmid

1 Environment Activation / Framework re-installation

You will use the **same** `conda` **environment** that you set up for assignment 1. However, we need to download a new version of the Python framework from MOODLE, and re-install the package it contains in this virtual `conda` environment.

- Activate the virtual environment you previously created:
  `$ conda activate py311_ai_assignments`
- For most shells, your command prompt should now have changed, to indicate that the virtual conda environment named `py311_ai_assignments` is now **active**.
- Download the file `ai_assignment4.zip` from the MOODLE course page.
- Unzip the `ai_assignment4.zip` into an empty directory.
- We will refer to this directory as your **base** directory.
- In your shell, navigate to the **base** directory.
- Issue the following command:
  `$ pip install -e .`
  (it will install the new `ai_assignments` package and its dependencies into your active `conda` environment)
- You are now ready to tackle the practical part of assignment 4!

2 Theoretical Questions (12 pts)

There is a separate quiz on MOODLE for theoretical questions, covering **propositional logic**, **theorem proving** using resolution and **decision trees**. For the theoretical questions you have an **unlimited number** of attempts, **but no feedback** whether or not your answers are correct.

3 ID3 Algorithm (Problem Instances + Code = 3 + 9 pts)

For the **practical** part of the assignment, you will be asked to implement the following decision tree learning algorithm that you already heard about in the lecture:

- ID3 Algorithm (`ai_assignments/decision_tree/id3.py`)

Please **only modify and upload the indicated source file (id3.py)**!

The practical assignment has a corresponding MOODLE quiz. This quiz contains questions, which contain training sets (similar to the problem instances in previous assignments). You will have to use this data to learn a decision tree using the ID3 algorithm.

Here is what to do for the ID3 Algorithm:

- Implement the ID3 algorithm in `ai_assignments/decision_tree/id3.py`. There are 3 TODOs in the source file that explain what you need to implement and what things to be careful about.
- Make sure your updated `conda` environment is active: (see instructions in Section 1)
  `$ conda activate py311_ai_assignments`
- Download the training sets (encoded as a JSON files) from MOODLE.
- In a shell, try:
  `$ python learn.py <training-set-from-quiz> id3`
- Copy and paste the solution hash to the respective Decision Tree question in the quiz.
- After you are done, upload your implementation as a **.py** file to MOODLE!

Visualisation & Debugging Hints

You can generate and visualise additional training sets and (optionally) the decision boundaries of the learned tree.

Caveat: Visualizing only works for training sets with exactly two features. The problem instances from MOODLE have more dimensions but feel free to generate training sets yourself (manually or using the provided script).

- Generate a training set with 20 examples:
  `python generate.py trainset 20 test/trainset.json`
- Visualise the data set:
  `python visualize_data.py test/trainset.json`
- Visualise the learned decision boundaries of the tree:
  `python visualize_data.py test/trainset.json --tree test/id3.pt`
- Visualise the learned decision boundaries of the tree (but only the first level):
  `python visualize_data.py test/trainset.json --tree test/id3.pt --depth 1`

In case you are stuck, and your algorithm behaves in weird ways:

- generate a small training instance like this one:

```
{"type": "TrainingSet",
  "X": [[1.0, 2.0],
        [3.0, 2.0]],
  "y": [0, 1]}
```

- compute decision tree by hand
- compare to the one your algorithm computed
- increase complexity (more samples, more complex feature space)