**Project 3 MDP**

**Xuzheng Lu**

**G34363475**

**GitHub:** <https://github.com/LeanderLXZ/mdp>

**1. Files description**

|  |  |
| --- | --- |
| mdp/  ├── doc.docx  ├── README.md  ├── input/  │   ├── i1.txt  │   │  │   └── i8.txt  ├── results/  │   ├── results\_1.csv  │   ├── results\_2.csv  │   └── results\_3.csv  └── src/  ├── value\_iteration.py  ├── policy\_iteration.py  ├── policy\_iteration\_linear.py  ├── notebook.ipynb  └── utils.py | // document file  // readme file  // input folder  // TXT files of inputs  // results folder  // CSV files of results  // srouce code folder  // value iteration  // policy iteration  // policy iteration based on linear equations  // jupyter notebook for experiments  // utilities |

**2. Arguments for algorithms**

|  |  |
| --- | --- |
| board\_file\_path | string, the path of input file |
| threshold | float, default=0.01, the threshold for stop the iteration |
| init\_policy\_direction | int, default=None, the index of the chosen direction in ['up', 'right', 'down', 'left'] for initializing the policy. If None, randomly assign directions to the initial policy. |
| improve\_p\_with\_v | Boolean, default=Flase, set True to improve the values while improving the policy |
| use\_arrow | Boolean, default=False, set True to use arrows for display |
| verbose | Boolean, default=False, set True to display extra information |

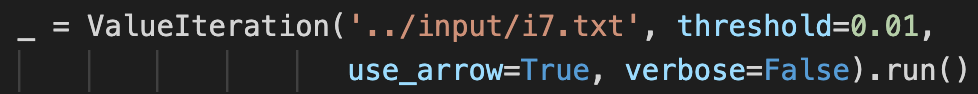
**3. Run the code**

**1) Run value iteration**

Command:

python value\_iteration.py

You can set the configuration of the algorithm ` value\_iteration.py `:



Output example:

A picture containing electronics, computer, circuit

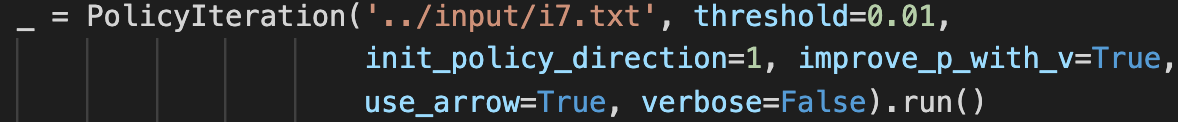
Description automatically generated

**2) Run policy iteration**

Command:

python policy\_iteration.py

You can set the configuration of the algorithm in ` policy\_iteration.py `:



Output example:

A picture containing electronics, computer, circuit

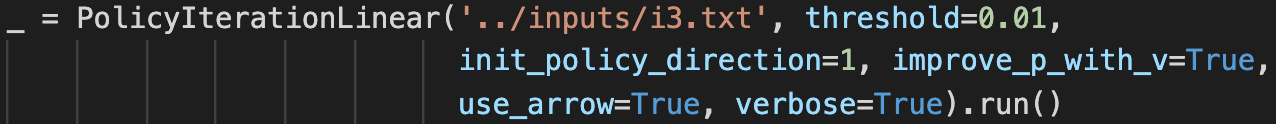
Description automatically generated

**3) Run policy iteration based on linear equations solving**

Command:

python policy\_iteration\_linear.py

You can set the configuration of the algorithm in ` policy\_iteration.py `:



Output example:

A screenshot of a cell phone

Description automatically generated

**4. Experiments (Runtime)**

**1) Value Iteration vs. Policy Iteration vs. Policy Iteration based on Linear Equations Solving**

**A screenshot of a cell phone

Description automatically generated**

**A close up of a logo

Description automatically generatedA screenshot of a cell phone

Description automatically generated**

**Conclusion:**

**In value iteration, the algorithm needs to check all of the possible actions. In policy iteration, the algorithm only needs to check one fixed action. Therefore, policy iteration**

**2) Compare different initialization method**

**A screenshot of a cell phone

Description automatically generated**

**A close up of a map

Description automatically generatedA close up of a map

Description automatically generated**

**Conclusion:**

**3) Improve the values while improving the policy**

**A screenshot of a cell phone

Description automatically generated**

**A close up of a map

Description automatically generatedA close up of a map

Description automatically generated**

**Conclusion:**