AI 533	**Intelligent Agents and Decision Making**	Winter 2025

OREGON STATE UNIVERSITY
School of Electrical Engineering and Computer Science
Instructor: Sandhya Saisubramanian
Submitted by: Manogna Challoju

Total points: 100 Mini Project 1 Due date: Feb 10, 2025

1. **Part A: Exact Methods for Solving MDPs (55 points)**

   1. **(15 points) Environment Setup**

      The environment is a 4x4 Gridworld where the agent starts at $(0, 0)$ and aims to reach the goal at $(3, 3)$. There are water cells at $(1, 1)$ and $(2, 2)$ getting a reward of -5, and wildfire cells at $(0, 3)$ and $(3, 0)$ gets a reward of -10. All other cells give a reward of -1, except the goal which gives $+100$. Transition dynamics include an 80% chance of moving in the intended direction and 10% chance of sliding to a neighboring cell.

   2. **(25 points) Value Iteration Implementation**

      Value iteration was implemented with discount factors $\gamma = 0.3$ and $\gamma = 0.95$. The results are:

      For $\gamma = 0.3$:

      - **Policy:**

        ```
        down    up      down    down
        left    down    right   down
        right   down    down    down
        right   right   right   G
        ```

      - **Value Function:**

$$\begin{bmatrix} -1.42 & -1.41 & -0.92 & 2.84 \\ -1.41 & 0.01 & 3.9 & 19.51 \\ -0.92 & 3.9 & 21.4 & 82.52 \\ 2.84 & 19.51 & 82.52 & 0 \end{bmatrix}$$

      For $\gamma = 0.95$:

      - **Policy:**

        ```
        down    right   down    down
        down    down    right   down
        right   down    down    down
        right   right   right   G
        ```

      - **Value Function:**

$$\begin{bmatrix} 64.1 & 69.02 & 74.58 & 80.55 \\ 69.02 & 75.61 & 81.93 & 89.1 \\ 74.58 & 81.93 & 89.86 & 97.17 \\ 80.55 & 89.1 & 97.17 & 0 \end{bmatrix}$$

Observation: With a higher discount factor ($\gamma = 0.95$), the agent prioritizes long-term rewards and avoids high-penalty cells more effectively compared to $\gamma = 0.3$.

3. **(15 points) Policy Iteration Implementation**

   Policy iteration was executed with $\gamma = 0.95$. The policy and value function are similar to the value iteration with $\gamma = 0.95$:

   Policy:

```
down    right   down    down
down    down    right   down
right   down    down    down
right   right   right   G
```

   Value Function:

$$\begin{bmatrix} 64.1 & 69.02 & 74.58 & 80.55 \\ 69.02 & 75.61 & 81.93 & 89.1 \\ 74.58 & 81.93 & 89.86 & 97.17 \\ 80.55 & 89.1 & 97.17 & 0 \end{bmatrix}$$

   Observation: The equivalence of results confirms that both methods converge to the same optimal policy and value function for $\gamma = 0.95$.

2. **Part B: Imitation Learning (45 points)**

   1. **(15 points) Policy Simulation**

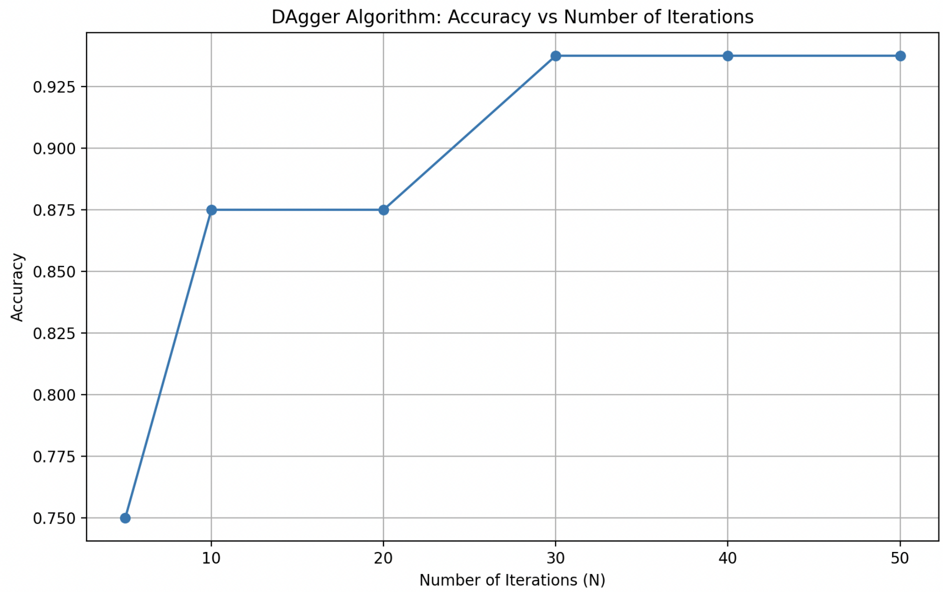
      An episode generated using the optimal policy is:

```
State: (0, 0), Action: right, Reward: -1
State: (0, 1), Action: right, Reward: -1
State: (0, 2), Action: right, Reward: -5
State: (0, 3), Action: down, Reward: -1
State: (1, 3), Action: down, Reward: -1
State: (2, 3), Action: down, Reward: 100
State: (3, 3), Action: None, Reward: 100
```

      Due to stochastic transitions, episodes vary in length and rewards.

   2. **(30 points) DAgger Algorithm Implementation**

      The DAgger algorithm was implemented using a Decision Tree classifier. I experimented with $N = \{5, 10, 20, 30, 40, 50\}$ iterations and calculated the accuracy based on the proportion of actions matching the expert policy.



Observation:

- Varies for every iteration because of random value. For one example shown the accuracy increased rapidly from $N = 5$ to $N = 30$, reaching approximately 94%.
- After $N = 30$, accuracy plateaued, indicating the model had sufficiently learned the expert policy.
- The increase in accuracy with N is due to more diverse and comprehensive datasets collected over iterations, helping the classifier to better approximate the expert's behavior.