Autonomous Drone Navigation - A Delivery System

A P Vaideeswaran
Tejadhith Sankar

Course Presentation under Dr. Mathukumalli Vidyasagar

25 November 2024

Problem Setup

- A drone has to deliver a parcel from the warehouse to the customer.
- There are fixed obstacles (buildings, dead-ends, and the layout boundary wall) at fixed locations.
- Find a policy that the drone can follow, given various simulation scenarios.

Simulation Modeling

- Model the interaction between the agent (Drone) and its environment by a Markov Decision Process (MDP).

Simulation Modeling

- Model the interaction between the agent (Drone) and its environment by a Markov Decision Process (MDP).
- The agent is characterized by its state.
- The states can either be discrete or continuous.

Simulation Modeling

- Model the interaction between the agent (Drone) and its environment by a Markov Decision Process (MDP).
- The agent is characterized by its state.
- The states can either be discrete or continuous.
- We consider **both** the possibilities
  - **Discrete:**
    - View the entire city as a grid
    - Each grid point is a state
    - **Possible actions:** stay, left, right, up, or down

Simulation Modeling

- Model the interaction between the agent (Drone) and its environment by a Markov Decision Process (MDP).
- The agent is characterized by its state.
- The states can either be discrete or continuous.
- We consider **both** the possibilities
  - **Discrete:**
    - View the entire city as a grid
    - Each grid point is a state
    - **Possible actions:** stay, left, right, up, or down
  - **Continuous:**
    - The agent need not occupy any particular grid point.
    - It can lie between grid points and have velocities in two directions.
    - Here, the speed limit is 5 m/s in either direction with **0** initial velocity.
    - **Possible actions:** Stay or accelerate (in 2D).
    - The acceleration is limited to 2 m/s^2 component-wise with a time-step of 0.5 s.

Reward Mechanism Deterministic

- In general, the rewards can either be probabilistic or deterministic.

Reward Mechanism Deterministic

- In general, the rewards can either be probabilistic or deterministic.
- Here, too, we consider both the scenarios:
- **Deterministic:**
  - Moving closer to the target $\Rightarrow$ small $+$ve reward
  - Moving away from target $\Rightarrow$ small -ve reward
  - Sometimes back-tracking is essential $\Rightarrow$ do not penalize heavily!

Reward Mechanism Deterministic

- In general, the rewards can either be probabilistic or deterministic.
- Here, too, we consider both the scenarios:
- **Deterministic:**
  - Moving closer to the target $\Rightarrow$ small +ve reward
  - Moving away from target $\Rightarrow$ small -ve reward
  - Sometimes back-tracking is essential $\Rightarrow$ do not penalize heavily!
  - Hitting an obstacle (denoted by '#' in the simulation) $\Rightarrow$ highly -ve reward
  - Reaching the target (denoted by 'T') $\Rightarrow$ highly +ve reward
  - In this case, staying is not an action.

Reward Mechanism Probabilistic

- **Probabilistic:**
  - Most of the reward system is *similar* to the previous case.
  - **For Discrete MDP**:
    - Each wind velocity component is a random number sampled from a Distribution (here $\mathcal{U}$[-1, 1]).
    - This random number is added to the reward for motion towards/away from the target.
    - A particular direction might be relatively encouraged/ discouraged.

Reward Mechanism Probabilistic

- **Probabilistic:**
  - Most of the reward system is *similar* to the previous case.
  - **For Discrete MDP**:
    - Each wind velocity component is a random number sampled from a Distribution (here $\mathcal{U}$[-1, 1]).
    - This random number is added to the reward for motion towards/away from the target.
    - A particular direction might be relatively encouraged/ discouraged.
  - **For Continuous MDP**:
    - Each wind acceleration component is a random number sampled from a Distribution (here $\mathcal{U}$[-1, 1]).

Reward Mechanism Probabilistic

- **Probabilistic:**
  - Most of the reward system is *similar* to the previous case.
  - **For Discrete MDP**:
    - Each wind velocity component is a random number sampled from a Distribution (here $\mathcal{U}$[-1, 1]).
    - This random number is added to the reward for motion towards/away from the target.
    - A particular direction might be relatively encouraged/ discouraged.
  - **For Continuous MDP**:
    - Each wind acceleration component is a random number sampled from a Distribution (here $\mathcal{U}$[-1, 1]).
  - In this case, "stay" is an action.
  - Depending on the wind, move in a particular direction later if need be.
  - A small negative reward is set.
  - In general, "staying" is inefficient. However, it *might* be required.

Reward Mechanism Probabilistic

- **Probabilistic:**
  - Most of the reward system is *similar* to the previous case.
  - **For Discrete MDP**:
    - Each wind velocity component is a random number sampled from a Distribution (here $\mathcal{U}$[-1, 1]).
    - This random number is added to the reward for motion towards/away from the target.
    - A particular direction might be relatively encouraged/ discouraged.
  - **For Continuous MDP**:
    - Each wind acceleration component is a random number sampled from a Distribution (here $\mathcal{U}$[-1, 1]).
  - In this case, "stay" is an action.
  - Depending on the wind, move in a particular direction later if need be.
  - A small negative reward is set.
  - In general, "staying" is inefficient. However, it *might* be required.
  - Presence of wind $\Leftrightarrow$ moving obstacles (minor collision/ push).

Simulation Specifics

- Generate layouts **randomly** given the drone and target locations and an obstacle density parameter.
  - Drone and target location must not coincide
  - Neither location must correspond to and beyond the boundary walls.

Simulation Specifics

- Generate layouts **randomly** given the drone and target locations and an obstacle density parameter.
  - Drone and target location must not coincide
  - Neither location must correspond to and beyond the boundary walls.
- Employ Depth First Search (DFS) to check layout solvability.
  - Meaningless to compare the performance of learning algorithms when the agent can **never** reach the target.

Simulation Specifics

- Generate layouts **randomly** given the drone and target locations and an obstacle density parameter.
  - Drone and target location must not coincide
  - Neither location must correspond to and beyond the boundary walls.
- Employ Depth First Search (DFS) to check layout solvability.
  - Meaningless to compare the performance of learning algorithms when the agent can **never** reach the target.
- Overall, we perform four simulation types:
  - Discrete MDP with Deterministic Reward
  - Discrete MDP with Probabilistic Reward
  - Continuous MDP with Deterministic Reward
  - Continuous MDP with Probabilistic Reward

Simulation Specifics

- Generate layouts **randomly** given the drone and target locations and an obstacle density parameter.
    - Drone and target location must not coincide
    - Neither location must correspond to and beyond the boundary walls.
- Employ Depth First Search (DFS) to check layout solvability.
    - Meaningless to compare the performance of learning algorithms when the agent can **never** reach the target.
- Overall, we perform four simulation types:
    - Discrete MDP with Deterministic Reward
    - Discrete MDP with Probabilistic Reward
    - Continuous MDP with Deterministic Reward
    - Continuous MDP with Probabilistic Reward
- In the Discrete MDP with Deterministic Reward case, we study the effect of "greedy" decision-making

Reward System Specifics

Action	Reward
completion	100000
hit obstacle	-100000
reduce distance	10
increase distance	-5
stay	-1

Table: Rewards for Different Actions

How are we assessing Learning Algorithms?

- The layout size is varied from 5 x 5 to 8 x 8 in steps of 1 (4 sizes).
- The obstacle density is varied from 0 to 0.4 in steps of 0.1 (5 cases).

How are we assessing Learning Algorithms?

- The layout size is varied from 5 x 5 to 8 x 8 in steps of 1 (4 sizes).
- The obstacle density is varied from 0 to 0.4 in steps of 0.1 (5 cases).
- 20 simulation scenarios common to each **simulation type** and **learning algorithm**.
- To assess the performance corresponding to each scenario:
  - The first case with the greatest obstacle density given a particular size is used for learning.
  - The remaining 50 + (51 x 3) cases are for policy assessment.
  - The average number of steps required is reported across the corresponding 50 cases.
  - Averaging reduces the effects of one-off poor/ excellent performances.

How are we assessing Learning Algorithms?

- The layout size is varied from 5 x 5 to 8 x 8 in steps of 1 (4 sizes).
- The obstacle density is varied from 0 to 0.4 in steps of 0.1 (5 cases).
- 20 simulation scenarios common to each **simulation type** and **learning algorithm**.
- To assess the performance corresponding to each scenario:
  - The first case with the greatest obstacle density given a particular size is used for learning.
  - The remaining 50 + (51 x 3) cases are for policy assessment.
  - The average number of steps required is reported across the corresponding 50 cases.
  - Averaging reduces the effects of one-off poor/ excellent performances.
- 4 learning algorithms are considered:
  - Proximal Policy Optimization (PPO) [both MDP types]
  - Deep-Q Network (DQN) [discrete MDP]
  - Advantage Actor-Critic (A2C) [both MDP types]
  - Soft Actor-Critic (SAC) [continuous MDP]

Coding Framework

- The gymnasium framework was used for all the simulations.
- All the learning algorithms are provided by the Stable-Baselines3 framework.
- Simulation Link

Effect of Greedy Decision Making

- Studied only for the first simulation type.

Effect of Greedy Decision Making

- Studied only for the first simulation type.
- The discount factor (γ) is set to 0.
- When present in state X_i, decide X_{i+1} by the prospective reward.
- If the prospective reward is the same for multiple states, choose one randomly.

Effect of Greedy Decision Making

- Studied only for the first simulation type.
- The discount factor (γ) is set to 0.
- When present in state X_i, decide X_{i+1} by the prospective reward.
- If the prospective reward is the same for multiple states, choose one randomly.
- **Advantage:** A simple model which solves the grid "quickly."
- **Disadvantage:** Sometimes, the location of the agent keeps oscillating
  - Not able to leave an "obstacle zone"
  - Requires significant backtracking!
  - Foresight helps to prevent such cases
  - In the presence of randomness like wind, foresight is indispensable.

Remarks

- The detailed simulation results can be viewed: **link**

Remarks

- The detailed simulation results can be viewed: **link**
- Similar parameters are chosen for all the environments.
- Algorithms might perform better, given algorithm-specific tuning.

Remarks

- The detailed simulation results can be viewed: **link**
- Similar parameters are chosen for all the environments.
- Algorithms might perform better, given algorithm-specific tuning.
- **Goal of the simulation:** Given a **level-playing field**, which algorithm does the best?
- Only the observations and insights are presented for the sake of brevity.

Observations and Insights

- **Discrete MDP with Deterministic Reward:**
  - As the size and/or density increase, the number of required iterations increases.
  - **Performance Order:** Greedy $>$ PPO $>$ DQN $>$ A2C.
  - The poor performance of A2C might be due to:
    - Poor exploration
    - Potential training instabilities
  - PPO might be performing very well due to:
    - Sound Exploration
    - comparable stability to trust-region methods
  - DQN has an intermediate performance.

Observations and Insights

- **Discrete MDP with Probabilistic Reward:**
  - Similar set of observations
  - Higher environmental complexity, higher required iterations.
  - **Performance Order:** PPO > DQN > A2C
  - DQN performance falls dramatically.

Observations and Insights

- **Discrete MDP with Probabilistic Reward:**
  - Similar set of observations
  - Higher environmental complexity, higher required iterations.
  - **Performance Order:** PPO $>$ DQN $>$ A2C
  - DQN performance falls dramatically.
- **Continuous MDP - Both Reward Mechanisms**
  - The required iterations are comparable across both reward mechanisms.
  - Performance Order PPO $>$ A2C $\geq$ SAC.
  - Thoughts behind the poor performances of PPO and A2C:
    - Deterministic rewards result in poorer exploration and sub-optimal policies.
    - The reward stochasticity seriously impedes convergence (probabilistic case).
    - The time steps for training were less (4000 - 19000).
    - More complex algorithms might require more time to learn.

Acknowledgments

We would like to thank Dr. Vidyasagar for providing us with this valuable opportunity to experiment with various RL algorithms, get practical insights, and learn an RL Framework.