AERO 489/689 Foundations of Aerospace Autonomy

Fall 2023

Programming Project

**Context:** In this project,we will continue with the autonomous rover example from programming assignment 2. You may reuse your code from Programming Assignment 2. The goal is to give our agent probabilistic reasoning capabilities so it can avoid storms, which was the primary reason for losing the last generation of Marscorp’s rovers. In addition, we will give the agent the ability to do some autonomous fault diagnostics. We provide the whole problem context below.

Consider an autonomous rover on the surface of Mars. In this scenario, the Mars rover is traversing a 6x6 grid. The goal of the agent is to find and retrieve a biological sample, and to return it to the starting position. The agent’s state is specified by its location (x,y) and its orientation (left, right, up, down).

There are certain hazards in the environment:

* Rocks (which destroy the rover on contact (they’re very sharp))
* Storms (which destroy the rover on contact (they’re very powerful), but also degrade the air quality in adjacent (not diagonally adjacent) cells)
* Sandy ground (which indicate *past* storms)

The agent can take only certain actions:

* Move forward
* Rotate clockwise/counterclockwise (CW/CCW)
* Drill (to obtain the biological sample in the correct cell)
* Spectrometer (to determine if the biological sample is present)
* Air quality (to determine if the air quality is degraded in a cell)
* Traction (to determine if a cell is sandy)
* Lidar (to determine if rocks are ahead and if so, how far)

Note: The true state of the environment is represented by the “hidden-cells” in the knowledge base. This includes storm locations which are *fixed*. Do not treat the hidden storm location as a random variable.

**Problem 1**

Since you used so many rovers on Programming Assignment 2, you only have second-hand rovers available for the Project. These rovers are more susceptible to faults, such as battery system faults, solar panels being covered in dust, and getting rocks stuck in their wheels. Luckily, you can deal with these faults by restarting the battery system, using your solar panel wipers, or auto-cleaning your wheels respectively. The only issue is that your only way to diagnose the faults is through three pieces of information: your comms signal, your movement speed, and detecting drift in your movements. Both a Battery System Fault (random variable B) and dust in the solar panels (S) can cause a low battery power alarm (L). When battery power is low, that results in a weak comms signal (WC) and slower movement (SM). Slow movement can also be caused by a wheel subsystem failure (W), which in turn is due to a rock stuck in the wheel (R). The wheel failure also results in movement drifts off-center (MD). The relevant conditional PMFs for these variables are provided below.

A table of numbers and letters

Description automatically generated with medium confidence

1. Plot/draw the Bayesian network and implement it using the code provided. **WIP**
2. Use the code to answer the following test queries: **DONE**
   1. Probability of solar panels covered in dust given slow movement
   2. Probability of a battery system fault given weak comms
   3. Probability of rock stuck in wheel given slow movement and low battery power.
3. Integrate this code into the agent’s code **WIP**
   1. Complete the generate\_fault\_detection\_bayes\_net function.
   2. Complete the get\_fault\_prediction function.
4. Report the % of wins with fault prediction vs without fault prediction capabilities. You can do this with Problem 2 complete or not, but please make clear which is the case, as you will have different victory %s.
5. [Grad students only]: Find a way to improve upon the get\_fault\_reaction function. Show that your solution is better than the current get\_fault\_reaction function, by shortening the elapsed sim time or increasing the number of victories.

**Problem 2**

Now that we know a bit more about Mars thanks to Programming Assignment 2, we can use our upgraded air quality sensors to determine if the particulate level, or ppm, is high, medium, or low. The combination of the air quality measurements you make along with the sandy ground measurements will help you construct a probabilistic inference model for storms. The Bayesian network for *one particular cell (2,2)* is shown in the following figure:

A diagram of a storm

Description automatically generated

So there is information on adjacent, and diagonally adjacent cells. The PMFs for the PPM measurements are given here:

A table with numbers and symbols

Description automatically generated

In the event that you are diagonal to one storm and adjacent to another, the adjacency probabilities will take precedence.

1. Your task is to implement the probabilistic reasoning portion of the code, which will help reduce the number of rovers lost to storms. That is, when exploring, you should use your storm probabilities to help you break ties between possible actions.
2. Completion of the compute\_storm\_probs function for Problem 2.
3. Completion of the generate\_bayes\_nets function for Problem 2.
4. Report the % of wins and compare it to the one from Programming Assignment 2.

**Deliverables:**

1. PDF document with detailed answers to all the questions above, including the relevant code snapshots.
2. Zip file with all of the code. Make sure that the code runs with only the files provided in the zip file.