CS238 Project Status Update

Group Members:
Levi Lian (levilian)
Suvir Mirchandani (smirchan)
Benjamin Newman (blnewman)

Introduction

In this project we are investigating whether or not an agent can learn a language by receiving instructions on how to achieve a goal. The setup is as follows: there is an agent in a room trying to get to a specified point. While the agent has a map of the room and knows where it is in the room, it does not know where the point it is trying to get to is. At every timestep, the agent receives an observation in the form of a (command, boolean) pair issued by a omniscient observer who knows where the agent and the goal point is. The command tells it one of four directions it should go (in the language it does not understood) and the boolean tells it if it was "correct" in its last movement (i.e. if its last movement took it closer to goal point). The agent uses this information to update its belief states as to a) where the goal point is and b) which command refers to which direction. It then chooses the next action to take. We are modeling this problem as a POMDP and plan on using `POMPDPs.jl` to examine solutions.

Problem Formalization

- **States**: The first two items on the list below are known to the agent, so for those items, the belief update for will have no probabilistic component.
  - Position p
  - Previous Command c_{t-1}
  - Command $\rightarrow$ Direction Mapping Function $C(cmd)$
  - Goal position G
- **Actions**: $a \in [0, 2\pi)$ represents the (continuous) direction in which the agent chooses to move.
- **Reward**: We have a -100 reward for bumping into the walls and a +20 reward for getting to the goal state.
- **Transitions**: Transitions in the problem are deterministic. If each action a causes a change in position Δp_a in the direction of the action, then $T(s' \mid s, a) = \delta_{s'}(p + \Delta p_a)$. We will implement this world as having a number of discrete tiles, so s', and p refer to the tiles associated with the positions.

- **Belief States**: Our belief states will be modeled as the following:
  - Position p: this is known to the agent, so it will just be a point (x, y)
  - Previous Command c_{t-1}: this is also known to the agent, so it will just be a number between 1 and 4
  - Command $\rightarrow$ Direction Mapping Function. This is not known to the agent explicitly, so we need to have some model over it. For each of the four commands the agent hears, it has a distribution over which direction, $(0, 2\pi]$, that command refers to. Because the actions it can take are continuous, this would have to be some kind of Dirichlet with infinitely many parameters, which does not really work, so instead we use a Gaussian Mixture Model and update the belief as follows:

    ```
    IF feedback = "YES":
        Add to distribution of previous command 𝒩(a, σ)
        and normalize
        Add to distributions of all other commands 𝒩(a + π, σ)
        and normalize
    IF feedback = "NO":
        Add to distribution of previous command 𝒩(a + π, σ)
        and normalize
        Add to distributions of all other commands 𝒩(a, σ)
        and normalize
    ```

  - Goal position. Because the agent gets feedback on whether or not an action takes it closer to the goal, we can update our belief over the goal position as follows:

    ```
    IF feedback = "YES":
        for all positions p′ in direction a + π:
            b(p′) = 0
    IF feedback = "NO":
        for all positions p′ in direction a:
            b(p′) = 0
    ```

We are currently implementing the necessary functions in `POMPDPs.jl` to solve this POMDP. Time permitting we will start to look into higher level modeling of the speaker's belief of what the agent's belief state is.

Revised Timeline

- 11/15/2018 - Submit Project Status Update
- 11/22/2018 - Have all necessary `POMPDPs.jl` interface functions defined

- 11/29/2018 - Fine tune visualization code
- 12/03/2018 - Finish analysis of POMDP solving methods
- 12/07/2018 - Final Project Paper due