Progress Report: Social Feedback for Robotic Collaboration

Emily Wu, Brown University

February 8, 2016

1 Introduction

In my proposal, I introduced a project to enable robots to interact socially with humans in human robot collaboration tasks. In order to establish this social feedback loop, robots must be able to both perceive and convey their mental state to the human partner. Presented below is what I have accomplished so far.

2 Implementing the Domain

For this project, I built a domain describing an object handover interaction on the BURLAP framework. In order to build a domain, the states, actions, observations, reward function, and their interactions must be described within the framework. I have implemented three different variants of these domains, and begun evaluation on them in a simulated environment.

2.1 Base implementation

In the basic implementation, we define the states, actions, transition function, reward function, and observations and observation models as follows:

States are given by a tuple of $\langle \omega, \mathcal{O}, b_H \rangle$. The object the human wishes the robot to handover is w, the set of all objects and their locations and names are given as $\mathcal{O}$, and the robot's belief about which object the human believes they will hand over is represented by a distribution b_H over $\mathcal{O}$.

The **actions** available for the robot to take are to `WAIT` (take no action) `PICK(x)` some object x, `ASK(x)` about some object(s) x, `POINT(x)` to some object, or `LOOK(x)` at some object x.

The **transition function** describes how state changes in response to the robot's actions. The desired object ω only changes once the robot has `PICK`ed the correct object. Similarly, $\mathcal{O}$ only changes if an object was picked up. B_h, the belief of the human about which object the robot hands over, changes with every action the robot takes. Specifically, b_H is updated in a bayesian manner as if the human had observed the robot's action. Specifically:

$$b_{Ht}(\omega_t) = p(a_t|\omega_t) \sum_{\omega_{t-1}} p(\omega_t|\omega_{t-1}) b_{Ht-1}(\omega_{t-1})$$

$p(a_t|\omega_t)$ gives the probability of observing action a_t (the robot's last action) given an object ω_t, which is the object the human believes the robot is trying to communicate. this directly mirrors the observation functions described below.

The **observations** that the robot can make of its environment (i.e., the human's actions) are speech and gesture made by humans. Speech is given by a NLP provided by google, while gesture is provided by a kinect.

The **observation function** describes how observations are produced by the current state. In a POMDP, the current state is hidden, so it is from these observations that the agent (robot) must determine the current state. Our observations pertain mostly to the desired object ω. The details of the observation function are unchanged from the proposal, but I use the same models to update the state variable b_H, except from the perspective receiving observations from the robot's actions.

The **reward function** describes how the robot receives reward according to its actions and the hidden states. This essentially incentivizes and disinsentivizes certain behaviors. In this domain, we provide a large positive reward for `PICK`ing the correct object, and a large negative reward for `PICK`ing the incorrect object. In addition, several smaller negative rewards are given for taking actions such as `ASK`ing the user questions or as a penaty for "annoying" the human.

2.2 Incremental Picks

On top of the base implementation, I have experimented with variations that improve the robot's ability to communicate. The first of these variations is the inclusion of an incremental pick action. In this variation, the robot must deliberately choose the pick action until it observes the pick is complete to