

Social Feedback for Robotic Collaboration

Emily Wu, Brown University

April 6, 2016

Contents

1	Abstract	2
2	Introduction	3
3	Related Work	3
4	Technical Approach	3
4.1	POMDP Overview	3
4.2	Model Description	4
4.3	POMDP Definition	4
4.3.1	Observation Function	4
4.3.2	Transition Function	5
4.4	Policy Generation	5
4.5	Systems	5
5	Evaluation	5
5.1	User Studies	5
6	Future Work	5

1 Abstract

2 Introduction

Collaboration is a process that relies heavily on communication for its success. When humans collaborate, this communication is both obvious and implicit—we not only instruct and request aid from each other, but we also tacitly monitor our partners for signs of approval and understanding while producing these signals ourselves. However, in robotics, many of these vital components of successful communication are missing or lost. The absence of these signals likely account for numerous failures in human-robot collaborative tasks. In order to give robots these missing communicative skills, we employ *social feedback* signals to provide human-like communication to both inform the human participant of the robot’s state and request information from the participant.

Feedback refers to the responses that are received by an agent when it takes some action. These feedback responses inform the agent about the success or failure of its actions. *Social feedback* therefore refers to social signals that convey this information. These can be explicit signals, such as “I want this object”, or implicit signals, like a perplexed expression. When humans interact with another human, they use this feedback to improve the flow and clarity of the collaboration, which in turn improves its success. For my thesis work, I have developed a framework that describes how robotic agent should likewise generate social feedback for its human partner in human-collaborative tasks, and show that it improves the speed and accuracy of human-robot interaction.

In this work, we address the object delivery task. In this task, a set of objects are laid out on a table within the robot’s reach. A human participant requests an object from the robot using speech and gesture (pointing). The robot must interpret the human’s speech and gesture and deliver the requested object to the human. The task then repeats with the remaining objects on the table. This task is achievable without social feedback; the robot need only wait until enough information is given and then deliver the correct object. However, we will show that by adding social feedback actions, such as asking questions, looking at objects, and pointing at objects, we will achieve better accuracy and speed as well as improved user experience.

We will solve this problem by formulating it as a Partially Observable Markov Decision Process (POMDP), which will allow us to dynamically and flexibly determine how to choose social feedback actions. First, we will describe a two agent model and use this to motivate the construction of a POMDP. We will next discuss how we solved this POMDP and other measures we took to allow our system to respond dynamically and fluidly.

To evaluate our approach, Stop. Hammertime.

3 Related Work

4 Technical Approach

4.1 POMDP Overview

Partially Observable Markov Decision Processes (POMDPs)^{citation} are used to model Markov Decision Processes^{citation} where the true state is not known. Instead, the agent receives observations that are generated by the true state, and must infer what the true state is from the observations. Thus, the agent maintains a belief over true states which is updated as it receives new observations. The agent must then use this belief state to determine which action to take to maximize its expected value of the reward over time. This splits the POMDP into two main components, a state estimator and a policy generator.

State Estimator The state estimator is given an initial belief and uses Bayesian mathematics to update its belief over time. In order to perform this update, the state estimator has a model of how the hidden states emit observations as well as how the state evolves in response to actions performed by the robot. The first model is described by the observation function and the second by the transition function.

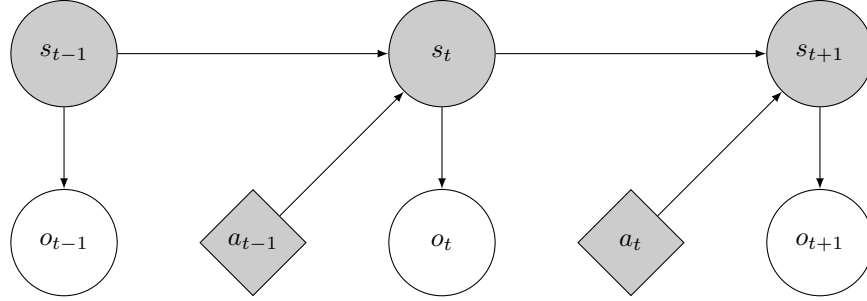
Policy Generator The policy generator is the means by which actions are chosen by the agent. Specifically, the policy generator should choose actions that maximize the agent’s expected reward over time. The reward for taking a particular action in a state is given by a reward function. These state-action pairs are called policies, and are generated by a POMDP Solver. Generally, POMDPs are solved by converting to their equivalent belief-MDP, which is an MDP where the state is distribution over the POMDP’s hidden states, a belief. In subsequent sections we will refer to this belief over states as b .

Formally, a POMDP is defined as a tuple $\{S, A, T, R, \Omega, O\}$, where :

- $s \in S$ is a hidden state.

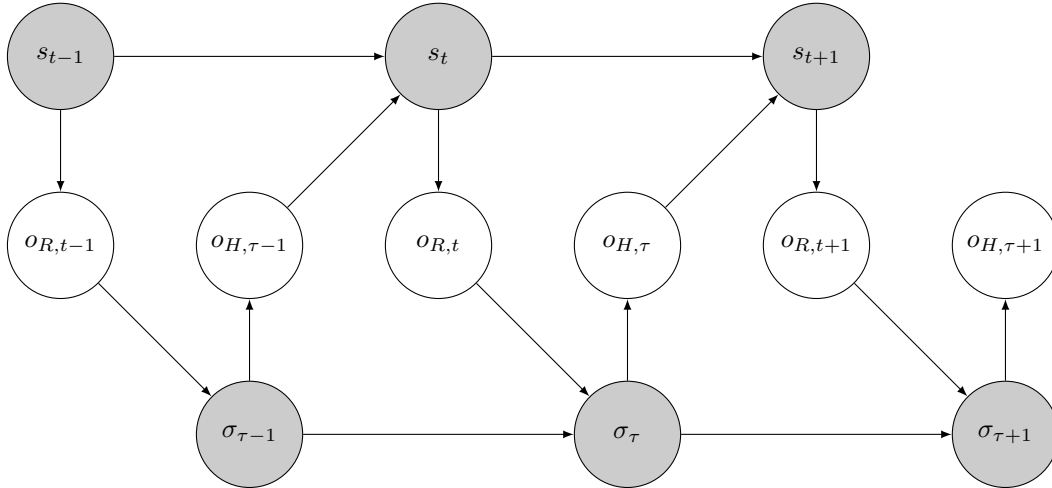
- $a \in A$ is an action that an action takes.
- $T : S \times A \times S \rightarrow \mathcal{R}$ is a transition function that returns the value $p(s'|s, a)$, the probability of transitioning to state $s' \in S$ from state $s \in S$ when taking action $a \in A$.
- $R : S \times A \rightarrow \mathcal{R}$ is a reward function that returns the reward for taking action $a \in A$ in state $s \in S$.
- $o \in \Omega$ is an observation.
- $O : S \times A \times O \rightarrow \mathcal{R}$ is an observation function that returns the value of $p(o|s, a)$, the probability of observing observation $o \in O$ when action $a \in A$ is taken from state $s \in S$.

A typical bayesian network for a POMDP is as follows.



4.2 Model Description

To model this human-robot interactive task, we will use a Partially Observable Markov Decision Process (POMDP) **citation**. POMDPs are used to model Markov Decision Processes (MDPs) **citation** where the true state is unknown, and observations must be made to perform a belief distribution over hidden states. In order to inform the construction of our POMDP model, we will examine a two-agent bayesian network.



In the above model, we represent robot states s_t , human states σ_τ , and observations generated by the robot $o_{R,t}$ and by the human $o_{H,t}$. Crucially, we observe that

4.3 POMDP Definition

4.3.1 Observation Function

According to our double-agent model, the human emits observations as though it were an agent interacting with our robotic agent. Thus, we choose an observation model that depends on the human's belief about the robot's state, β . Specifically, the human will choose an action according to its estimate that the robot will hand them that object. In order to define this function, we will first have to define a base-level observation function.

Base-Level Observation Function The base level observation function describes the probability of an observation conditioned only on the object: $p(o|\iota)$. For our object delivery domain, we will define two base-level observations, one for language and one for gesture.

Speech Model: Language is interpreted according to a unigram gesture model. An utterance l is broken down into individual words, $w \in l$:

$$p(l|\iota) = \prod_{w \in l} p(w|\iota) = \prod_{w \in l} \frac{\text{count}(w, \iota.\text{vocab})}{|\iota.\text{vocab}|} \quad (1)$$

where $\text{count}(w, \iota.\text{vocab})$ is the number of times word w appears in ι 's vocabulary.

Gesture Model: All gestures are interpreted as a straight armed point. These pointing gestures are selected from a normal distribution centered at the object's location.

Define the angle between the vector defined by the pointing gesture and the vector from the human's arm to the object ι to be θ_ι . The probability of a particular gesture is then

$$p(g|\iota) = \mathcal{N}(\theta_\iota|0, v) \quad (2)$$

where v is a hand-tuned variance.

Posterior Observation Function We will use the base-level observation functions defined above to define a posterior observation that considers the effects of the base level observation function. Specifically, the human chooses an observation proportional to the robot's belief in the desired object if the human had chosen that observation, i.e.,

$$p(o|s) = p(o|\iota, \beta) \triangleq \eta p(\iota|o)_\beta \quad (3)$$

$$= \eta \frac{p(o|\iota)p(\iota)}{\sum_{\iota'} p(o|\iota')p(\iota')} \quad (4)$$

Toy Example

4.3.2 Transition Function

4.4 Policy Generation

4.5 Systems

5 Evaluation

5.1 User Studies

6 Future Work