This is the first in a series of blog posts on either new or important ideas in multi-agent learning. There really isn't much out there trying to put some of these ideas into easily understood prose and consequently a lot of key innovations are locked up in fairly inaccessible papers. My goal here is to make it easier to catch up on the field and to highlight some key advances that have been happening recently.

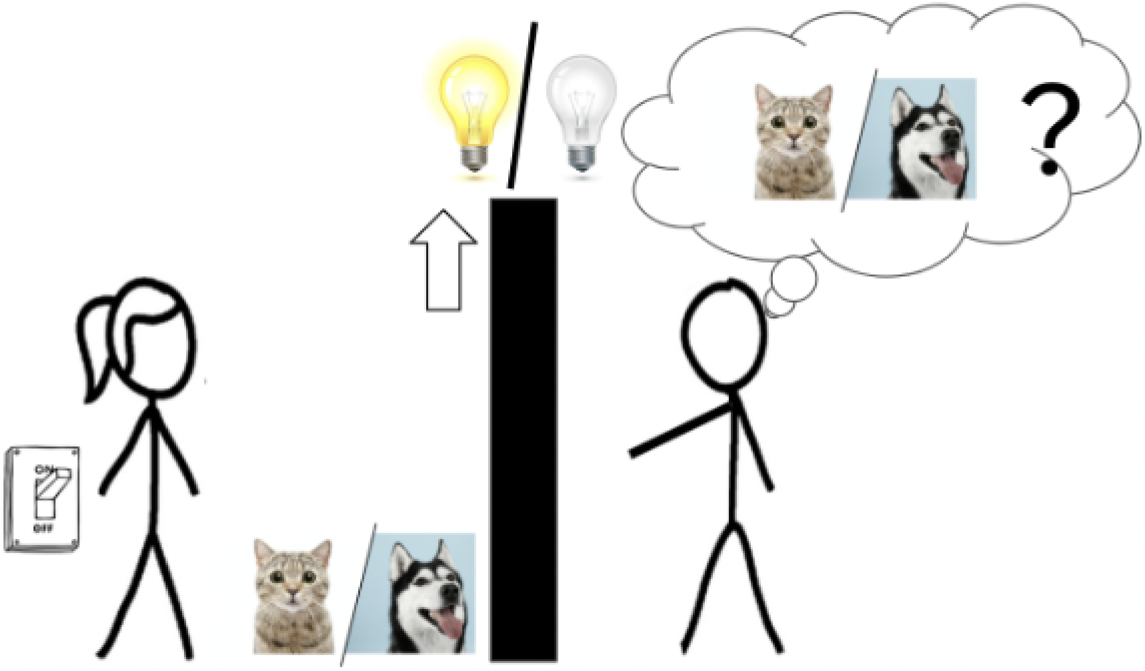
TLDR:

- If we're coordinating with an unknown agent, it's useful to treat it as though it doesn't know any conventions and instead is just interpreting only grounded information it observes.
- Off-belief learning gives us a way to generate agents that are grounded / convention-free and assume that the agents they are playing with are convention-free as well.
- More generally, off-belief learning lets us build agents that assume their observations are coming from playing with an arbitrary agent of our choosing and that their actions in turn will be interpreted by an arbitrary other agent.

Introduction

We're starting with what I think is one of the biggest innovations to come out in the last year or two: Off-Belief Learning (**OBL**). **OBL** is kind of like the blockchain in that a lot of questions I ask are often answered with "pretty sure off-belief learning solves this" but it's different than the blockchain because it's obviously useful. It's a clever solution to a very ubiquitous problem in multi-agent learning.

First, the problem **OBL** is looking to tackle is the problem of improving *zero-shot coordination* by removing *arbitrary convention formation* from multi-agent self-play (I'll define what each of these pieces mean, don't worry). First, arbitrary convention formation. There's a nice example of this in the paper:



Here Alice is trying to communicate whether she has a cat or a dog to Bob who will try to guess which she has. They both get +10 points if he gets it right and -10 to both if he gets it wrong. She can also pay 5 points to remove the wall, clearly exposing which pet she has or flip a light bulb to its on or off state. Finally, Alice can choose to exit from the game for a reward of 0.5 and Bob can exit the game for a reward of 1 instead of guessing. If we run self-play (i.e. Alice and Bob play this game together hundreds of times), Alice and Bob will learn a convention like "light-bulb on means dog, light-bulb off means cat" and will use the light switch instead of removing the wall.

That's all fine and good for Alice and Bob but what if we suddenly swap out Bob with another player, Candace? Candace has no idea about the convention; when Alice flicks the lighbulb Candace will think "I have no idea what that means. I'm going to exit the game for a reward of 1." It's a fine strategy for playing with Bob, but it's not a good strategy for playing with an arbitrary, new player. This is the challenge of *zero-shot coordination*: how can we construct a strategy that should perform well no matter who our partner is?

Note, this isn't a super well-defined problem, in principle no such strategy exists.
For example, even if Alice were to remove the wall, it's possible that Bob could look at the revealed dog and exclaim "Cat!" Under the pessimal assumption that Bob is unable to distinguish cats from dogs, the optimal thing for Alice to do is to simply exit the game. We need to make some assumptions for coordination to be possible at all. One nice assumption that you could make on a partner is that Bob is an *optimal grounded agent* (**OGA**): an agent that acts optimally based on everything it observes but does not assume

that observed actions are indicative of any conventions (this is a loose definition, we'll give a more technical one later). If Bob is an OGA agent and Alice flips the switch, Bob will simply exit the game because they don't know what the switch means. On the other hand, if Alice opens the wall, Bob will immediately guess the right pet.

In other words, an **OGA** is an agent that acts only based on its observations of the world, but **ignores** the actions of its partner except to the extent that they reveal information about the world i.e. optimally grounded Bob ignores the light switch but pays attention to the pets when they're revealed.

More intuition on optimal grounded agents

Okay, so how do we construct a self-play process that returns grounded agents instead of agents with arbitrary conventions? One of the key insights of the OBL paper is defining such an agent to be one that acts as though "all prior actions came from a random agent and that all my actions will be interpreted by an agent that believes all actions came from a random agent." Lets unpack that a bit because it's deeply unintuitive. If we have such an agent, then looking at the history of actions we have seen so far, we can't try to read conventions out of them since we *believe* they came from a random agent. However, for our action we want to signal as much information as possible to our partner since we *believe* that our partner is a copy of ourselves and will extract as much information as possible from the actions without assuming any conventions.

Lets filter this through the lens of Alice and Bob. For any action Alice takes, she knows that Bob will believe her action was taken randomly. If she flips the light switch, he won't know what to do with that information and will exit the game. If she opens the door, it doesn't matter to Bob that the door opening occured randomly, he knows what the pet is and will guess it right.

Building an optimal grounded agent

Lets move up one layer of abstraction and try to define this in technical terms. We're going to be slightly more general for a second and then restrict things to refer back to our **OGA**.
First, some notation. First, we're going to assume for simplicity that we're playing a turn-based game that has some max horizon H. We're also going to assume that all actions are observable by all agents. None of these assumptions are strictly necessary, but they make the exposition a lot easier.

We're working with partially observed systems here so instead of the MDP state s the agents are going to operate on action observation histories (**AOH**) $\tau^i = (o_1, a_1, \dots, a_{t-1}, o_t)$ where o_j is the agents partial observation of the true state s_t at time-step j. As an example, in poker the true world state s would be the concatenation of all the players cards and the agent observation for agent i , o^i would be all the cards visible to agent i. We'll also have the fully-observed historical trajectory $\tau = (s_1, a_1, \ldots, a_{t-1}, s_t)$. Finally, we'll use π^i to represent the policy of agent i and π to denote the joint policy of all the agents. Since we'll want to distinguish between different joint policies, we will also use subscripts like π_0 or π_1 to distinguish between different joint policies.

First, lets define the notion of *counterfactual value*. This game is partially observed so based on the trajectory agent i has seen so far, τ^i it's going to have some belief over the true trajectory τ. We denote this as $\mathcal{B}_\pi(\tau|\tau_i) = P(\tau|\tau_i, \pi)$ i.e. the probability of being in trajectory τ given that we've seen trajectory τ_i and we're playing joint policy π. Note the conditioning on joint policy π, the belief assumes we know who we're playing with.

The *counterfactual value* is defined as

$$V^{\pi_0 \to \pi_1}(\tau^i) = E_{\tau \sim B_{\pi_0}(\tau|\tau_i)}\left[V^{\pi_1}(\tau)\right]$$

Similarly, we can define a counterfactual Q-value, the value of seeing an observation and taking a particular action as

$$Q^{\pi_0 \to \pi_1}(a|\tau^i) = \sum_{\tau_i, \tau_{i+1}} B_{\pi_0}(\tau_t, \tau_t^i)\left[r(s_t, a) + \mathcal{T}(\tau_{i+1}, \tau_i) V^{\pi_1}(\tau_{i+1})\right]$$

We can then define the **OBL operator** which takes in an initial policy π_0 and converts it into a new policy π_1 via:

$$\pi_1(a|\tau^i) = \frac{\exp\left(Q^{\pi_0 \to \pi_1}(a|\tau^i)/T\right)}{\sum_{a'} \exp\left(Q^{\pi_0 \to \pi_1}(a'|\tau^i)/T\right)}$$

The **OBL operator** just returns the softmax of the counterfactual Q values with a temperature T.
Key takeaway / theorem: As $T \to 0$, the OBL operator returns the optimal grounded policy as long as π_0 is a policy that does not condition on the actions.

High level intuition on the OBL operator

The *counterfactual value* is the expected value of trajectory τ^i if we've been playing with π_0 up till now and are going to be playing with π_1 afterwards. An agent that optimizes this quantity is going to interpret the trajectory it sees as coming from π_0 but then play afterwards as though it is playing as part of joint policy π_1.

Now, lets connect this back to the **OGA**. Suppose that π_0 is a totally random agent (or any other convention-free agent i.e. an agent that conditions only on observations but not the actions of its partner) and we've currently observed trajectory τ^i. Then, there's no value in trying to read a convention out of τ^i, just extract all the information about the true state of the world you can without assuming any convention. Now, we know as much as we can but what action should we take? Well, if we assume that in the future our actions are going to be interpreted as though they're coming from a random agent, we might as well signal as much information about the true state of the world as we can.

Actually solving for the OBL policy

We have our OBL operator, we now need an RL procedure that returns the optimum.

Option 1:

We can perform standard Bellman iteration. If we expand the counterfactual Q-values slightly, we can write them in a form that's amenable to sampling:

$$Q^{\pi_0 \to \pi_1}(a|\tau^i) = \mathbb{E}_{\tau_t, \tau_{t+k}} \left[\sum_{t'=t}^{t+k-1} r(s'_t, a'_t) + \sum_{a_{t+k}} \pi_1(a_{t+k}|\tau^i_{t+k}) Q^{\pi_0 \to \pi_1}(a_{t+k}|\tau^i_{t+k}) \right]$$

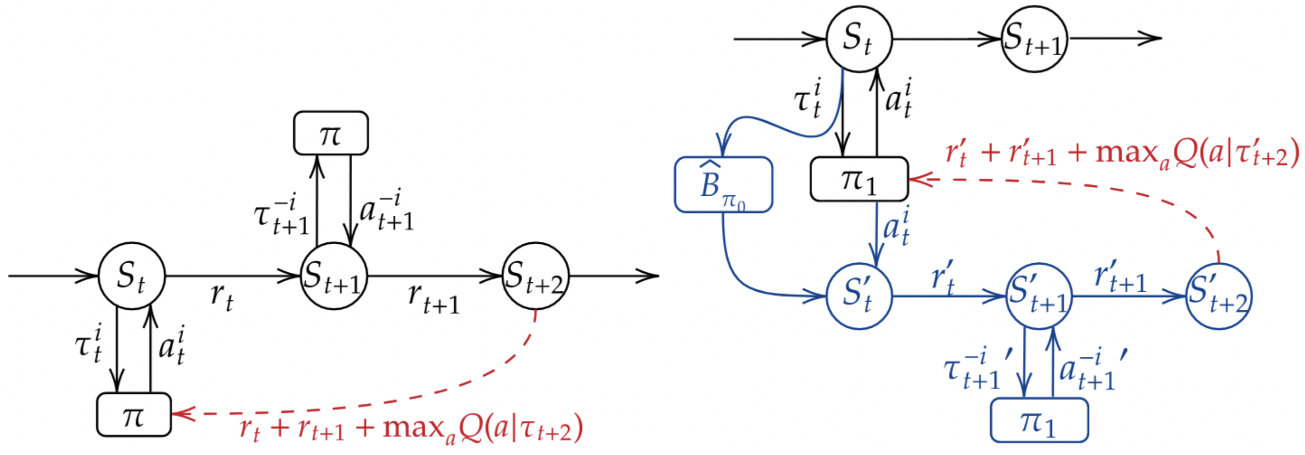
A quick note, the term $\mathbb{E}_{\tau_t, \tau_{t+k}}$ refers to the fact that our current player might not play again till k steps later. This is just an expanded Bellman equation for the counterfactual Q function and we can try to converge to a fixed point by drawing samples from this distribution and minimizing the difference between the LHS and RHS as we do in Q-learning. They call this approach **Q-OBL**.

Option 2

There's one serious challenge though: drawing samples. Samples from this distribution correspond to playing π_0 up till time t, and our current policy π_1 thereafter and these two policies might reach very different state distributions. For example, if π_0 is a uniform random policy, it might reach any interesting states with very low probability.

Here they introduce another trick to let them sample interesting states with higher probability, a method they call **LB-OBL**: Learned Belief OBL. We're going to play forwards under π_1 and then at each state we reach, we're going to take the partially observed trajectory we've seen under π_1, τ^i, and sample a trajectory τ that is consistent with τ^i but would have come from a random policy. In practice, this means we're training a belief model $B_{\pi_0}(\tau|\tau^i)$ and drawing samples from it (for more info on this, check out the section on learned belief models). We'll then take this new fictitious sample τ and use it to roll out a fictitious transition along it and compute the counterfactual Bellman update along that fictitious transition.

The paper provides a fairly helpful diagram of these two approaches with **Q-OBL** on the left and **LB-OBL** on the right.



How well does it work?

Really well! You can read the paper for that since the goal here was to get the ideas across.

Quick aside: learned belief models

The idea of using a learned belief model is fairly neat so here's a little section diving into it more, with a full exposition in the paper Learned Belief Search: Efficiently Improving Policies in Partially Observable Sethttps://arxiv.org/pdf/2106.09086.pdftings. For partially observable games, it is useful to have a persistent estimate of the likely true state of the world. Sometimes, we can explicitly enumerate this true state of the world but in general it can be fairly high dimensional. If we train a model that predicts from the current partially observed **AOH** a distribution over the true state of the world, we can simply sample from it for all sorts of tasks; in the linked paper they use it for search.

However, the true state of the world, which combines all private agent observations + agent actions + public knowledge can be very high dimensional. Here we can apply a useful decomposition of a game into the set of common knowledge (what everyone knows) and private knowledge (what only you know). For many games, the set of private knowledge can be fairly small! For example, in 2-player poker games like Texas Hold-em, the private knowledge is simply the two cards that the opponent holds. Our belief model can simply draw samples of the private knowledge and combine this with the public knowledge to form the complete state of the game.

In short, we are going to draw samples from our game and train a supervised model for our agents by taking the public observations and using them to predict the private observations of the agents. Since we're training this model on rollouts, we have the private observations and can use them as labels.