Semi-supervised reinforcement learning

Paul Christiano Follow
May 6, 2016 · 6 min read

Semi-supervised RL is similar to traditional episodic RL, but there are two kinds of episodes:

- “labelled” episodes, which are just like traditional episodes,
- “unlabelled” episodes, where the agent does not get to see its rewards.

As usual, our goal is to quickly learn a policy which receives a high reward per episode. There are two natural flavors of semi-supervised RL:

- Random labels: each episode is labelled with some fixed probability
- Active learning: the agent can request feedback on its performance in any episode. The goal is to be economical both with feedback requests and total training time.

We can apply a traditional RL algorithm to the semi-supervised setting by simply ignoring all of the unlabelled episodes. This will generally result in very slow learning. The interesting challenge is to learn efficiently from the unlabelled episodes.

I think that semi-supervised RL is a valuable ingredient for AI control, as well as an interesting research problem in reinforcement learning.

Applications and motivation

Application to AI control: expensive reward functions

As a simple example, consider an RL system which learns from the user pressing a “reward button”—each time the agent performs a task well the user presses the button to let it know. (A realistic design would more likely use verbal approval, more subtle cues, or performance measures that don't involve the user at all. But a very simple example makes the point clear.)

If our system is a competent RL agent maximizing button presses, it will eventually learn to deceive and manipulate the user into pressing the button, or to simply press the button itself.

We'd prefer that the system treated the button presses as *information* about what behavior is good, rather than the *definition* of what is good. Deceiving the user simply destroys the usefulness of that information.

This can be captured in the semi-supervised RL framework. Suppose that we have some expensive "ground truth" procedure that can reliably assess how good a system's behavior really was. We can use this procedure to define the reward signal in a semi-supervised RL problem. The agent can then use the reward button presses to learn effectively from the "unlabelled" episodes, after recognizing that button presses provide useful information about the ground truth.

Of course designing such a ground truth is itself a serious challenge. But designing an expensive objective seems much easier than designing a cheap one, and handling expensive objectives seems key to building efficient aligned AI systems. Moreover, if we are freed to use an expensive ground truth, we can rely on extensive counterfactual oversight, including bootstrapping, opening up a promising family of solutions to the control problem.

If we have good algorithms for semi-supervised RL, then the expensiveness of the ground truth procedure won't cause problems. The feedback efficiency of our semi-supervised RL algorithm determines just how expensive the ground truth can feasibly be.

Semi-supervised RL as an RL problem

Even setting aside AI control, semi-supervised RL is an interesting challenge problem for reinforcement learning. It provides a different angle on understanding the efficiency of reinforcement learning algorithms, and a different yardstick by which to measure progress towards "human-like" learning.

Methods for semi-supervised RL are also likely to be useful for handling sparsity and variance in reward signals more generally. Even if we are only interested in RL problems with full supervision, these are key difficulties. Isolating them in a simple environment can help us understand possible solutions.

Application to AI control: facilitating present work

In the short term, I think that counterfactual oversight and bootstrapping are worth exploring experimentally. Both involve optimizing an expensive ground truth, and so performing interesting experiments is already bottlenecked on competent semi-supervised RL.

Application to AI control: measuring success

Several AI control problems arise naturally in the context of semi-supervised RL:

- **Detecting context changes**. An agent's estimate of rewards may become inaccurate in a new context—for example, once the agent learns to perform a new kind of action. A successful agent for active semi-supervised RL must learn to recognize possible changes and query for feedback in response to those changes.
- **Handling uncertainty about the reward function**. An efficient semi-supervised RL agent must behave well even when it doesn't have a precise estimate for the reward function. In particular, it will sometimes have a much better and more stable model of the environment dynamics than of the reward function. This problem is especially interesting if we track performance during training rather than just measuring time to convergence.
- **Eliciting information and communicating**. In many environments acquiring information about the unobserved reward may require behaving strategically. For example, an agent might ask questions of a human overseer, in order to efficiently learn about what the overseer *would* decide if they performed an extensive evaluation. The agent is motivated to communicate effectively so that the overseer can quickly reach accurate conclusions. This is a key behavior for aligned AI systems.

We can study these problems in a semi-supervised RL setup, where we have a precisely defined objective and can easily measure success. Having a clean framework for measuring performance may help close the gap between problems in AI control and traditional research problems in AI.

Implementation

Experiments

The obvious way to study semi-supervised RL is to try and do it:

1. Start with any classic RL environment. For example, OpenAI recently published an awesome library here. The Atari, MuJoCo, and Classic Control environments are especially appropriate.
2. Rather than providing the rewards to the learner in every episode, provide them only when the learner makes a request (for example, have env.feedback() return the sequence of rewards in the most recent episode)
3. Measure performance as a function of both #episodes and #labels. For example measure performance as a function of (N + 1000F), where N is the number of episodes and F is the number of episodes on which feedback is requested.

These modifications are easy to make to the standard RL setup with just a few lines of code.

Algorithms

It’s easy to come up with some plausible approaches to semi-supervised RL. For example:

- Train a model to estimate the total reward of an episode, and use it to estimate the payoff of unlabelled episodes or to reduce variance of the normalized feedback estimator.
- Combine the above with traditional semi-supervised learning, to more quickly learn the reward estimator.
- Use the observations of the transition function in unlabelled episodes to make more accurate Bellman updates.
- Learn an estimator for the per-step reward.
- Use model-based RL and train the model with data from unlabelled episodes.

I expect the first strategy to be the simplest thing to get working. It will of course work especially well in environments where the cost function is easy to estimate from the environment. For example, in Atari games we could learn to read the score directly from the screen (and this is closer to how a human would learn to play the game).

More interesting experiments

Even very simple examples are interesting from the perspective of AI control, but more complex environments would be more interesting:

- Training an agent when the most effective reward estimator is manipulable. For example, consider a game where moving your character next to the score display appears to increase the score but has no effect on the actual score. We would like to train an agent not to bother modifying the displayed score.
- Training an agent to use a reward estimator which requires effort to observe. For example, consider a game that only displays the score when the game is paused.
- Using human feedback to define a reward function that has good but imperfect estimators. For example, we could teach an agent to play Pac-Man but to eat the power pellets as late as possible or to spend as much time as possible near the red ghost.
- Providing a sequence of increasingly accurate (and increasingly expensive) reward estimators. I think this is the most natural approach to generalizing from sloppy evaluations to detailed evaluations.

Conclusion

I think that semi-supervised is an unusually tractable and interesting problem in AI control, and is also a natural problem in reinforcement learning. There are simple experiments to do now and a range of promising approaches to try. Even simple experiments are interesting from a control perspective, and there are natural directions to scale them up to more compelling demonstrations and feasibility tests.

(This research was supported as part of the Future of Life Institute FLI-RFP-AI1 program, grant #2015–143898.)