IRL and VOI

Paul Christiano Follow
Sep 30, 2015 · 5 min read

Consider the following straightforward algorithm based on inverse reinforcement learning:

- Given your observations of the user so far, find the reward function that best explains their behavior (perhaps with some regularization/prior, or margin requirement, etc.)
- Take the action that maximizes this reward function (or optimize a weighted sum of reward functions, etc.)

When learning and interacting are interleaved, this algorithm can behave badly.

Why care?

In many settings learning and acting can be separated into disjoint phases: first an expert demonstrates a desired behavior, and then an agent implements it. So why care about whether we can mix the two?

One reason to care is interaction: we would like AI systems to query their users when they are uncertain, and to tell the user what they do and do not understand. But in order to learn from such interactions, agents must mix learning and acting.

A problematic example

Consider a simple robot tasked with delivering an item from a store. The robot is unsure about whether the user would prefer the item be delivered to their home or to their office. Based on previous observations the robot has a 60% credence that the item should be delivered to home, and a 40% probability that it should be delivered to work. For simplicity, assume that the predicted reward is either 0 or 1, and doesn't depend on anything other than where the item is delivered. The robot is using the simple algorithm above, so will try to deliver the item to the user's home.

Fortunately, one of the robot's routes home passes by the restaurant where the user is currently eating lunch. If the user sees the robot

delivering the item to their house, but the user actually wants the item delivered to their office, they can just say so—the robot is smart enough to make the appropriate inference.

Ideally our robot would actively seek out the user's input. But the simple algorithm above actually does the opposite.

If the robot accurately models its own future behavior, it will be willing to e.g. take a costly detour in order to avoid feedback. After all, it is currently maximizing a reward function that would prefer the item be delivered to the house, and feedback may cause it to bring the item to the office instead, reducing its payoff according to its current reward function.

(If the robot ignores the user's feedback, it will continue to deliver the item to the house as much as the user protests—and if user might turn it off, the robot will again take a detour to avoid the risk. If the robot ignores the fact that its values may change in the future, it's easy to construct other cases where it will go around in circles forever achieving nothing—and at any rate, it still won't actively seek out information, which is what we really want.)

Similar scenarios appear for a wide range of frameworks and are a traditional concern for researchers interested in AI safety per se. See in particular here.

Fixes

I know of two qualitatively different approaches to this problem. As far as I can tell both are viable, and at least one is necessary. The thing I find most interesting is that the two approaches seem so different.

Preferences about process

If we asked the user "is it a good idea for the robot to avoid you so that you don't have an opportunity to correct its behavior?" it's easy to predict that they would say "no." The robot could learn these kinds of "preferences" in the same way that it learns preferences over outcomes.

At this point, the thing the robot is learning is not human values per se, since in fact the human has no fundamental interest in the process that the robot uses. Instead the "values" that are learnt play the role of robust instructions.

This is the solution adopted by an approval-directed architecture, or an agent that mimics its user's behavior. It's worth noting that e.g. calculations about value of information can still appear in the agent's reasoning, because those considerations enter into the user's view about what the agent should do.

Indirect normativity

Alternatively, we can give the robot a static indirect definition of its preferences rather than having its preferences change over time. In this set up, only the robot's empirical beliefs change over time, and "what states of affairs are valuable" is treated as an ordinary empirical question.

Probably the most natural approach is to choose a joint distribution over sequences of observations and the utility function of the human user (such joint distributions are necessarily already explicit or implicit in any value learning setup). Following Daniel Dewey, we could then define the utility function "the expected utility given by the posterior over utility functions, conditioned on all of the robot's interactions." (Critically, this includes future observations—if we include only past observations this is exactly the same as the naive procedure.)

This setup introduces a correlation between the agents' observations and its preferences. This correlation will influence the plans generated by a sophisticated planning algorithm, and recovers the correct treatment of value of information—in fact exactly the same treatment that the planning algorithm uses for the value of empirical information.

I think that this solution is much more intuitively attractive than the last one, and in simple domains it is relatively easy to implement. But in the general case, where the robot cannot actually compute the posterior over utility functions, it is somewhat more complicated and much more philosophically subtle. This approach forces us into the Bayesian framework and raises tricky issues about preference aggregation / normative uncertainty. The first approach seems to be more ad hoc and less principled, but is very easy to adapt to bounded reasoners, limited memory, and approximate inference.

Conclusion

I don't think that this is a serious obstacle for value-learning based approaches to AI control. I do think that it's an issue worth thinking

about, if only to make our models and discussion more precise, and to date I haven't seen it considered explicitly. I think that implicit disagreements about this kind of detail may be lurking in the background of many discussions about superintelligence.

I have become much more optimistic about AI control since I started taking the first approach more seriously—not just to this problem, but to a wide range of similar difficulties. I feel even more optimistic with two reasonable options on the table.

(Thanks to Owain Evans and Andreas Stuhlmüller for useful discussion and comments.)