Delegating to a mixed crowd

Paul Christiano Follow

Dec 15, 2014 · 3 min read

Suppose I have ten programs, each a human-level agent. I suspect that at least one or two of them will behave helpfully, but the others may behave maliciously. How can I get useful work out of the combination?

Here's a very rough proposal:

- Each agent receives an initial budget of points.
- By default, each agent runs slowly in an independent environment. An agent may pay points to receive extra computational resources.
- At any time, an agent may propose an action and claim that I would approve of it—perhaps it's an instruction to give to a robot, a message to display to me, or a modification to this top-level loop. They also estimate how much I'd be willing to pay them for the time they spent thinking up the action.
- Any other agent can challenge the proposed action, or the estimate for how much I'd be willing to pay. If so, the actor and challenger have an argument about it, which I arbitrate.
- Running arguments is the key ingredient of the proposal: I need to arbitrate arguments which are potentially (1) extremely frequent, (2) extremely complicated, and (3) between extremely smart people. And I want to do it without spending almost any of my time. I think that an idea along these lines might work, where agents wager points about what we will believe if we hear them out, and we concentrate our attention on claims where there is significant disagreement.
- If there are no successful challenges, the action is taken and the actor is compensated. If there is a successful challenge, the action is not taken and the challenger is compensated instead, at the expense of the actor (we can use a similar argument to determine the extent of the compensation). In addition to these payments, many points may be won or lost over the couse of the argument itself.

Would this work? I don't know. I'd guess that it would work iff the arguments worked; if the arguments work correctly and the malicious agents try to significantly interfere with the helpful agents, the helpful agents will probably be able to extract points from them.

Odds and ends

Details

This game is still underspecified; many details would have to be filled in, and many tweaks and elaborations would be required to get something sensible.

We can use the system itself to address the question "how should the system work?" For example, if the honest agent realizes that the setup is subject to manipulation, she can immediately recommend a change to fix the problem. As long as her initial budget is large enough to convince us that we should change the system, the problem can correct itself.

Why not go straight for the throat?

The most important issue would be understanding which of the agents actually want to be helpful and why. If the argument system is working well, it might be used to address this problem directly; and once we knew which agent actually wanted to be helpful, we could get rid of the others.

So why do I care about the mixed-crowd setting at all?

1. I think that an implementation of the transform (helpful + ?) → (helpful) may be a useful ingredient in building (helpful) to start with. For example, we might apply similar ideas at a lower level or iterate the proposal to "amplify" slightly helpful agents into very helpful agents.

2. This procedure can be used to define a computation (where the "actions" are all computational steps) which "I would approve of" as a method to determine which agents want to be helpful. Defining a process to determine X is generally easier than determining X directly, so this places lower demands on the argument system.

3. Determining what agents want to be helpful may prove to be a particularly nasty problem. It may be that none of the agents reliably want to be helpful across contexts, or that there is no

easy way to see that any agent is motivated to be helpful (even if it is), or so on. We might want to have helpful behavior in the meantime.

Delegating to different crowds

In principle we could also delegate to agents who only care about collecting as many points as they can. But in the self-interested case, collusion becomes a significant concern.

We could apply the same idea to delegate to people. The first step would be understanding whether and how the argument scheme can work well enough (with people) to support an application like this.