Reliability amplification

Paul Christiano Follow
Oct 20, 2016 · 9 min read

In a recent post I talked about capability amplification, a putative procedure that turns a large number of fast weak agents into a slower, stronger agent.

If we do this in a naive way, it will decrease reliability. For example, if…

- Our weak policy fails with probability 1%.
- In order to implement a strong policy we combine 10 decisions made by weak agents.
- If any of these 10 decisions is bad, then so is the combination.

…then the combination will be bad with 10% probability.

Although the combination can be more powerful than any individual decision, in this case it is much *less* reliable. If we repeat policy amplification several times, our failure probability could quickly approach 1, even if it started out being exponentially small.

Complementary to capability amplification is *reliability amplification*: given a policy H that usually works, can we implement a policy H that works significantly more reliably?

To be slightly less imprecise (but still quite crude):

- Given a distribution **A** over policies that ε-close to a benign policy for some $\varepsilon \ll 1$, can we implement a distribution $\mathbf{A}^+$ over policies which is δ-close to a benign policy of similar capability, for some $\delta \ll \varepsilon$?

When I say that **A** is ε-close to a benign policy, I mean that there is a benign distribution over policies which has total variation distance at most ε from **A**. We can view this is as a simple mixture between a single aligned policy that is chosen with probability $(1-\varepsilon)$ and an adversarial policy that is chosen with probability ε.

I will move freely between talking about “policies” and “distributions over policies;” we can turn a distribution into a single policy by averaging.

The notion of “implement” here is exactly the same as for capability amplification: we can use any number of implementations of **A** and whatever other resources are at our disposal, and must compute **A⁺**’s responses to a sequence of observations, within some time limit such as a day.

Reliability amplification via deliberation

I think there is one very natural way to do reliability amplification—if some variant of this idea doesn’t work, I’m not sure what else to try.

As a warm-up, let’s consider the setting where the agents want to make a single decision rather than a sequence of decisions—we’ll return to the sequential case soon.

We sample three agents **A**, **B**, and **C** from the given distribution. We ask them to talk for a while and then each propose an action. Then we ask them to do approval voting amongst the three options, and randomly pick from the maximally-approved-of proposals.

If all three agents are aligned, then this proposal is clearly going to produce an aligned action. If two the agents behave adversarially then we are clearly out of luck, but that happens only with probability O(ε^2), so we can basically ignore it.

The interesting case is when there is one adversarial agent and two aligned agents. If we threw up our hands and picked randomly in this case, then we would have a failure probability of ε. So the question is: can we do better than that?

The obvious hope is that aligned agents can do better-than-chance at recognizing each other / at recognizing good actions. If they can get any non-negligible advantage, then we can get a non-negligible reduction in error probability (and by iterating the procedure we can potentially get large reductions).

The sequential case

So far we have talked about making a single decision. The problem is more challenging when we need to make a sequence of decisions.

We can try to simply generalize the voting approach, running a new vote for each action. To see the difficulty, suppose that the optimal policy looks as follows:

1. Generate some hard-to-transfer internal state X. Think of X as something like an intuitive vision for how to approach the situation.
2. Use X to generate a first action.
3. After making an observation, use X to generate a second action. It is important that the same state X be used in steps [2] and [3].

Now suppose that in the first step the three agents all follow this protocol correctly. They arrive at three different proposed actions, which are backed up by three different visions. Now we have to pick one of these actions, effectively committing ourselves to a single agent. This destroys our ability to ensemble effectively in future steps: if the chosen agent fails then we are out of luck, since the other two agents don't have the vision needed to follow-through on that action.

To get around this problem we need to count on a stronger property: whatever kind of private state X an agent could generate, the agents need to be able to jointly develop a "consensus" private state which serves the same function. Then if one agent fails, the other agents can also use the consensus private state in order to continue acting intelligently.

(It would also be OK if only pairs of agents could establish such a consensus state, as long as the third agent was able to arbitrate between the two in the event of a future failure.)

Implementation

As of commit 1cbaae, this approval-voting scheme is part of my ALBA implementation on GitHub.

It is implemented in amplify.reliability.amplify_reliability, and is applied as part of amplify.amplify directly prior to applying HCH.

As a result I've moved one issue from the "FIXME: will fail" column to the "TODO: needs improvement" column.

Three differences with redundant computation

There is an easy recipe for fault-tolerant computation: compute the result 3 times, and take the majority vote. This increases the computational cost by a factor of 3, but decreases the error rate from ε to $O(\varepsilon^2)$.

Our reliability amplification scheme is considerably more subtle, and if we were to formalize the reliability amplification problem as stated we would find it impossible. This section touches on the most important additional subtleties.

Building consensus

Redundant computation normally relies on the existence of a unique correct answer on which all uncorrupted computations will agree. In our setting, there may be a range of "correct" behaviors. This introduces the additional challenge of being able to agree on a particular correct answer (and to build a consensus version of the hidden state underlying that answer).

Without some assumption of this form, reliability amplification is impossible. For example, suppose that there are 99 possible good answers and 1 bad answer. Our distribution **A** picks at random amongst the 100 possible answers. If the copies of **A** with good answers are completely unable to agree with each other, then there may simply be no way to tell the good answers from the bad answer. In this case, we can never reduce the error probability below 1%.

Balanced abilities

In order for our scheme to work we need to assume that the abilities of all of the agents are roughly balanced. I think this is probably necessary for any scheme to work. If you have an ε chance of a malicious superintelligence and a (1−ε) chance of a friendly human, you are probably out of luck—in a deliberative process involving two humans and a superintelligence, the superintelligence is likely to have disproportionately much influence over the outcome even if it is wrong.

Our particular scheme fails outright in the context of powerful adversaries. In general there is a 3ε chance that *at least one* of the agents is unaligned. We may be able to prevent the unaligned agent from causing the group to adopt a really bad answer. But even in the best case, the unaligned agent will probably be able to tilt the balance between the two options proposed by the aligned agents, allowing them to increase the probability of failure. If the unaligned agent is no smarter than the aligned agents, then the aligned agents can counteract this effect by preferentially picking the better action. But if the unaligned agent is much smarter than the aligned agents, then it can increase the probability of failures which are too subtle for the unaligned agents to anticipate, and the aligned agents have no way to decrease the probability of these failures.

Less time to think

Traditionally redundant computation increases costs by a constant factor. But in the context of reliability amplification, we also make life harder for each agent: in addition to picking an action, they need to spend time arguing about actions and critiquing each others' choices. This gives each agent less time to think about their action. It is not clear whether we can offset this effect

Relation to capability amplification and ALBA

Necessity

Reliability amplification seems completely necessary if we want to do something like ALBA using imitation or imitation+RL, since otherwise reliability will fall with each iteration of capability amplification. Of course we could have a "robust" capability amplification procedure which does not decrease reliability. But designing such a procedure is strictly harder than reliability amplification. So I think it makes a lot of sense to split up the problem into two pieces.

If working with approval-direction and pure RL, there is actually a form of reliability amplification already baked in: if the overseer fails with probability 1%, then this only changes the reward function by 0.01, and an RL agent should still avoid highly undesirable actions. That said, capability amplification may still be necessary in a pure RL setup if we can't solve the RL problem to arbitrary precision. In that case we may always have some non-negligible probability of making a bad decision, and after capability amplification this probability could become too large.

Balancing capability/reliability

Reliability amplification decreases our agent's capability but increases its reliability. Capability amplification increases capability and decreases reliability.

The hope is that we can somehow put these pieces together in a way that ends up increasing both reliability and capability.

If our reliability amplification step achieves a superlinear reduction in error probability from ε to $o(\varepsilon)$, and our capability amplification causes a linear increase from ε to $\Theta(\varepsilon)$, then this seems almost guaranteed to work.

To see this, consider the capability decrease from reliability amplification. We know that for large enough N, N iterations of capability amplification will more than offset this capability decrease. This N is a constant which is independent of the initial error rate ε, and hence the total effect of N iterations is to increase the error rate to $\Theta(\varepsilon)$. For sufficiently small ε, this is more than offset by the $\varepsilon \to o(\varepsilon)$ reliability improvement from reliability amplification. So for sufficiently small ε we can increase both reliability and capability.

A reduction from ε to $O(\varepsilon^2)$ is basically the "best case" for reliability amplification, corresponding to the situation where two aligned agents can always reach correct consensus. In general, aligned agents will have some imperfect ability to reach consensus and to correctly detect bad proposals from a malicious agent. In this setting, we are more likely to have an $\varepsilon \to O(\varepsilon)$ reduction. Hopefully the constant can be very good.

There are also lower bounds on the achievable reliability ε derived from the reliability of the human and of our learning procedures.

So in fact reliability amplification will increase reliability by some factor R and decrease capability by some increment Δ, while capability amplification decreases reliability by some factor R' and increases capability by some increment Δ'. Our hope is that there exists some capability amplification procedure with $\Delta'/\log(R') > \Delta/\log(R)$, and which is efficient enough to be used as a reward function for semi-supervised RL.

I think that this condition is quite plausible but definitely not a sure thing; I'll say more about this question in future posts.

Conclusion

A large computation is almost guaranteed to experience some errors. This poses no challenge for the theory of computing because those errors can be corrected: by computing redundantly we can achieve arbitrarily low error rates, and so we can assume that even arbitrarily large computations are essentially perfectly reliable.

A long deliberative process is similarly guaranteed to experience periodic errors. Hopefully, it is possible to use a similar kind of redundancy in order to correct these errors. This question is substantially more subtle in this case: we can still use a majority vote, but here the space of options is very large and so we need the additional step of having the correct computations negotiate a consensus.

If this kind of reliability amplification can work, then I think that capability amplification is a plausible strategy for aligned learning. If reliability amplification doesn’t work well, then cascading failures could well be a fatal problem for attempts to define a powerful aligned agent as a composition of weak aligned agents.