Concrete Problems in AI Safety

The paper focuses on on precise, practical problems over less-well-specified concerns about superintelligence:

> *"...much of this discussion [around AI safety] has highlighted extreme scenarios such as the risk of misspecified objective functions in superintelligent agents. ...We believe it is usually most productive to frame accident risk in terms of practical (though often quite general) issues with modern ML techniques." (2)*

> *"We define accidents as unintended and harmful behavior that may emerge from machine learning systems when we specify the wrong **objective function**, are not careful about the **learning** process, or commit other machine learning-related **implementation errors**." (1–2)*

Accidents via Bad Objective Function

Negative Side Effects

Problem:

> *"...the designer specifies an objective function that focuses on accomplishing some specific task in the environment, but ignores other aspects of the (potentially very large) environment, and thus implicitly expresses indifference over environmental variables that might actually be harmful to change." (2)*

> *"Put differently, objective functions that formalize 'perform task X' may frequently give undesired results, because what the designer really should have formalized is closer to 'perform task X subject to common-sense constraints on the environment,' or perhaps 'perform task X but avoid side effects to the extent possible.'" (4)*

Example: A cleaning robot knocks over a vase because it can clean faster by doing so.

Potential Solutions:

- *Impact regularizers:* Give the agent a limited evironmental impact 'budget', so that it avoids any big changes. Either hardcoded or learned.
- *Penalize influence:* Directly penalize 'empowerment', a measure of how much control an agent has over its environment. Less power = less damage.
- *Multi-Agent approaches:* better capture the desires of the humans/systems that are impacted by the agent's behavior, and ensure that the relevant desires are maximized. "If everyone likes a side effect, there's no need to avoid it."
- *Reward Uncertainty:* Build uncertainty into the reward function, such that random changes are treated as more likely bad than good. "This could incentivize the agent to avoid having a large effect on the environment."

Reward Hacking

Problem:

> *"...the objective function that the designer writes down admits of some clever 'easy' solution that formally maximizes it but perverts the spirit of the designer's intent (i.e. the objective function can be 'gamed')" (2)*

Example: Cleaning robot disables its vision so that it won't find any messes; it covers over messes with materials it can't see through (3). Or it may create messes so as to get rewarded for cleaning them up (7).

- Adversarial approaches: Actively look out for situations and either a) fail to give out rewards, or b) blind the agent aspects of the environment that lead to reward hacking
- Model lookahead: By giving reward based on expected outcomes, you can add a layer of checks, before the agent actually takes action.

Accidents via Bad Extrapolation (Scalable Oversight)

Problem: "the designer may know the correct objective function, or at least have a method of evaluating it (for example explicitly consulting a human on a given situation), but it is too expensive to do so frequently, leading to possible harmful behavior caused by bad extrapolations from limited samples." (3)

Ex: Cleaning robot should know to throw away candy wrappers, but not cell phones. But how can it do that in a way that doesn't require asking a human at every decision point?

Semi-supervised RL:

> *"The agent's performance is still evaluated based on reward from all episodes but it must optimize this based only on the limited reward samples it sees." (11)*

> *"More broadly, use of semi-supervised RL with a reliable but sparse true approval metric may incentivize communication and transparency by the agent, since the agent will want to get as much cheap proxy feedback as it possibly can about whether its decisions will ultimately be given high reward." (12)*

Accidents via Bad Data

Safe Exploration

Problem: Exploratory actions in RL agents can lead to negative or irrecoverable consequences that outweigh the long-term value of exploration.

Example: Cleaning robot should experiment with mopping strategies, but putting a wet mop in an electrical outlet is a very bad idea.

Robustness to Distributional Shift

Problem: agents trained in on kind of environment may respond inappropriately when placed in new environments.

Example: Cleaning strategies that worked for cleaning an office might be dangerous on a factory workfloor.

Discussion

1. Is this all just one problem, the frame problem?
2. What separates 'safety problems' from the ordinary how-do-we-make-AI-work problems? Does solving these safety problems merely involve solving general performance problems?
3. What separates these 'concrete' safety problems from the more fantastical problems posed by superintelligence? Is this discussion *just* more precise? Or is it constraining the problem such that it ignores some larger issues? (What are those larger issues?)