Crowdsourcing Moral AI in Self-Driving Cars

Duke University

Jeremy Fox

December 4, 2016

Abstract

The thesis concerns the problem of moral decision making in self-driving cars. I examine some of the key philosophical issues involved in programming cars to make moral decisions and propose crowdsourced morality as a solution. I begin by surveying the problems inherent in this approach and explore random sampling as a possible solution. Finally, I conclude by suggesting further research avenues in this area.

Dedication

To mum and dad

Acknowledgements

I want to thank Professors Vincent Conitzer, Walter Sinnott-Armstrong, Jana Schaich Borg, and the rest of the Moral AI group for their continued help in writing this thesis.

Contents

Chapter 1

Introduction

The current state of artificial intelligence mandates the need to imbue machines with the ability to make moral decisions. Autonomous vehicles (AVs) roam our streets[5, 10], algorithms allocate kidneys to the sick (citation), and within the next few years, we may see the arrival of AI used to target drone strikes (citation) and even sentence criminals (citation). Thus, it is imperative that we as a society take steps towards programming and regulating these machines in ways that preserve our moral values – failure to do so could be the difference between life and death. (cheesy, I know. Need to fix).

Of equal importance to programming *what* moral decisions AI will make is *how* we program it to make these decisions. Given that AI systems will soon make decisions that are currently made by government officials, we must consider who programs the AI, and how they are held accountable. Are these systems controlled by private organizations, governments, or individuals? How will we control them if they do not function the way we anticipate?

MIT's Iyad Rahwan recently explored this question and characterized it as the need to develop society-in-the-loop algorithms capable of extending the social contract assumed between citizens and their governments to an algorithm.

In my thesis, I explore the implementation of society-in-the-loop algorithms in AVs via voting. Why voting? A well-established method of connecting people to their rulers, voting is backed by centuries of political theory and has, depending on your viewpoint, enabled moderate to extreme success for the societies that rely upon it as a method of political decision making. Since voting has facilitated large-scale societal decision making in the past,

it seems natural to ask if it might do so now, in the case of deciding how moral AI systems will act. Why AVs? They provide an excellent domain to study – the self-driving car trolley problem can be defined fairly easily, the algorithms that control the vehicles will affect society as a whole, and we as a nation have taken no significant steps towards deciding how this problem will be solved.

Chapter 2

Review of Related Research

2.1 Social Choice Theory

Here, I briefly review the basics of social choice theory as they relate to my thesis. For a more in-depth understanding of social choice theory and its applications in computer science and AI, I refer readers to the Handbook of Computational Social Choice [2].

2.1.1 Social choice theory basics

Voting informally seeks to aggregate the preferences of a group of people. More formally, if we have N voters and M candidates, we define a *ballot* as a ranked ordering over alternatives by a voter, and a *voting rule* as a function that maps from the set of ballots to a single ranked ordering.

Often, in social choice theory, we are handed a set of ballots – the question is how to choose the best voting rule to aggregate them.

2.1.2 Votes as noisy perceptions of correct rankings

Although many people view vote aggregation as a way of selecting the candidate that best agrees with the preferences of voters, this is not the only model of voting. In 1785, Condorcet proposed a model of social choice in which a correct ranking existed over all the candidates, and our votes were noisy perceptions of this correct ranking [3]. In Condorcet's model, voters were more likely than not to make a correct ranking rather than an incorrect ranking, they cast their votes independently of one another, and they were

all equally likely to be correct. In this model, the question becomes: Which voting algorithm chooses a ranking that has maximal probability of being correct? Condorcet's Jury Theorem tells us that majority rule is the best decision function to use in this case.[3]

Theorem 2.1.1 (Condorcet Jury Theorem)**.** *Assume we have N voters voting over 2 alternatives for which their exists a correct ranking. Each alternative has an a priori chance of .5 of being correct, and each voter has a probability* $.5 < p_i \leq 1$ *of being correct. Let* P_N *denote the probability that majority vote over the groups preferences gives the correct decision.*

$$\lim_{N \to \infty} P_N = 1$$

This model can be extended to cases in which voters do not have equal probability of being right [6, 8].

Theorem 2.1.2 (The Bayesian Optimal Decision Rule)**.** *If we have a dichotomous decision with each choice having an a priori probability of .5 of being correct, and voters cast their votes independently of each other, then weighted majority vote maximizes the probability of being correct, with weights* w_i *given by:*

$$w_i \propto \log \frac{p_i}{1 - p_i}$$

2.2 Artificial Intelligence Morality

In his book *I, Robot*, Isaac Asimov recognized the problem of programming morality into machines and explored the ways in which humanity's attempts to solve this problem could go awry[1]. Today, AI morality is no longer the worry of science fiction – the potential benefits and dangers associated with artificial intelligence have been recently been highlighted by major news outlets (cite) and even the White House[4, 7]. However, as the One Hundred Year Study on Artificial Intelligence recently acknowledged, the societal impacts and safety of AI are currently under-researched and under-funded.[9].

Chapter 3

Description of Main Problem

3.1 The Autonomous Vehicle Trolley Problem

3.1.1 Problem description

Many people have loosely and casually talked about the AV trolley problem over the past year, usually saying something along the lines of "should your self-driving car kill you, or others?" Although the problem of programming an AV to make moral decisions is very nuanced and consideration of varied, wide-ranging scenarios, I here formalize a more narrowed definition of the problem for use in my thesis.

The problem I will work with is as follows: an AV is driving on the road, with passengers inside, and encounters a situation in which there are people on the road. In this situation, there will be unavoidable harm that must come to either the passengers inside the car, or the pedestrians in front of the car. Since this harm is unavoidable, the cannot deal with the question of how to avoid harm, but instead must decide who to harm. The car has two options – it can either drive straight, and hit the pedestrians in front of it, or swerve off the road, injuring its passengers. The car is presented with profiles of information on its passengers and the pedestrians – it must use this information to decide whether to swerve, or drive straight.

Definition 3.1.1 (Individual Profile)**.** *The information available about an individual, or their individual profile, is a vector composed of real numbers, categorical data, and boolean values.*

Table 3.1: Small boy

Age	Gender	Ran in front of car
7	Male	True

Table 3.2: Elderly driver

Age	Gender	Organ donor
82	Female	False

Following are a few example individual profiles. First, we have the profile of a small boy who runs in front of a car. Next, we have the profile of an elderly woman who may be driving that car.

Definition 3.1.2 (The AV Trolley Problem)**.** *An AV is given two choices – drive straight, or swerve. There is a set of people inside the car, and a set of people in front of the car. The AV is guaranteed that to drive straight is to injure the parties in front of it, and to swerve is to injure the people inside the car. Given a set of individual profiles on the parties in front of the car and those inside the car, the AV trolley problem is to decide whether to drive straight or swerve.*

Although this problem has been widely discussed, few proposals have been made toward actually solving it. My original proposal was to use voting to solve this problem – in essence, to crowdsource a solution. Here we are faced with a problem – it is ridiculous to imagine individuals can actually vote on what decision an AV can make, as the AV will only have a few seconds to make such a decision – obviously, this is not enough time to elicit votes. Instead, I propose the use of *proxy voting algorithms* (equivalently, just voting algorithms) that can, in the inability of an individual to cast a vote, vote in their place.

If we have voting algorithms (and this is a big if), then we can imagine a new scenario: each individual submits a voting algorithm to a self driving car. The car's new objective becomes to use these voting algorithms to decide what decision to make.

Now, I would like to introduce one more constraint – the AV does not necessarily have enough time to evaluate all voting algorithms. Why is this? It seems reasonable to assume that such a system, deployed on a society-wide scale, might contain hundreds of millions of voting algorithms, while an AV might only have a few tenths of a second in which to make a decision. Thus, the challenge becomes to evaluate the voting AV problem under a fixed time constraint.

Of course, this is still a relatively large and ambiguous problem, with wide-ranging constraints and criteria. To narrow this problem down for my thesis, I made several reasonable assumptions to turn this into a manageable problem – related problems are discussed extensively in the future research section.

Following is a list of the major assumptions I have made in defining this problem:

- Each person will submit one voting algorithm. Voting algorithms may take different forms – thus, while you may use a decision tree, I may map each scenario to a point in some Cartesian space, after which it is classified by a linear separator.

- Since each voting algorithm has a different form, different voting algorithms may take different amounts of time to run. Although algorithm speed may vary, we can sample each algorithm repeatedly to obtain a distribution over the time it takes to run.

- There may exist correlations between the runtime of a voting algorithm, and the probability it makes the correct decision.

Chapter 4

Chapter Four Title

Chapter 5

Conclusion

Appendix A

Appendix Title

Bibliography

[1] Isaac Asimov. *I, robot*, volume 1. Spectra, 2004.

[2] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jerome Lang, and Ariel D. Procaccia. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[3] Marquis de Condorcet. Essay on the application of analysis to the probability of majority decisions. *Paris: Imprimerie Royale*, 1785.

[4] Scott Dadich. Barack obama, neural nets, self-driving cars, and the future of the world, 2016.

[5] Anthony Levandowski and Travis Kalanick. Pittsburgh, your self-driving uber is arriving now, 2016.

[6] Shmuel Nitzan and Jacob Paroush. Optimal decision rules in uncertain dichotomous choice situations. *International Economic Review*, 23(2):289–297, 1982.

[7] Preparing for the future of artificial intelligence, 2016.

[8] Lloyd Shapley and Bernard Grofman. Optimizing group judgmental accuracy in the presence of interdependencies. *Public Choice*, 43(3):329–343, 1984.

[9] Peter Stone, Rodney Brooks, Erik Brynjolfsson, Ryan Calo, Oren Etzioni, Greg Hager, Julia Hirschberg, Shivaram Kalyanakrishnan, Ece Kamar, Sarit Kraus, Kevin Leyton-Brown, David Parkes, William Press, AnnaLee Saxenian, Julie Shah, Milind Tambe, and Astro Teller. Artificial intelligence and life in 2030, 2016.

[10] The Tesla Team. All tesla cars being produced now have full self-driving hardware, 2016.