The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

A Voting-Based System for Ethical Decision Making

Ritesh Noothigattu
Machine Learning Dept.
Carnegie Mellon University

Snehalkumar 'Neil' S. Gaikwad
The Media Lab
Massachusetts Institute of Technology

Edmond Awad
The Media Lab
Massachusetts Institute of Technology

Sohan Dsouza
The Media Lab
MIT

Iyad Rahwan
The Media Lab
MIT

Pradeep Ravikumar
Machine Learning Dept.
CMU

Ariel D. Procaccia
Computer Science Dept.
CMU

Abstract

We present a general approach to automating ethical decisions, drawing on machine learning and computational social choice. In a nutshell, we propose to *learn* a model of societal preferences, and, when faced with a specific ethical dilemma at runtime, efficiently *aggregate* those preferences to identify a desirable choice. We provide a concrete algorithm that instantiates our approach; some of its crucial steps are informed by a new theory of *swap-dominance efficient* voting rules. Finally, we implement and evaluate a system for ethical decision making in the autonomous vehicle domain, using preference data collected from 1.3 million people through the Moral Machine website.

1 Introduction

The problem of ethical decision making, which has long been a grand challenge for AI (Wallach and Allen 2008), has recently caught the public imagination. Perhaps its best-known manifestation is a modern variant of the classic *trolley problem* (Jarvis Thomson 1985): An autonomous vehicle has a brake failure, leading to an accident with inevitably tragic consequences; due to the vehicle's superior perception and computation capabilities, it can make an informed decision. Should it stay its course and hit a wall, killing its three passengers, one of whom is a young girl? Or swerve and kill a male athlete and his dog, who are crossing the street on a red light? A notable paper by Bonnefon, Shariff, and Rahwan (2016) has shed some light on how people address such questions, and even former US President Barack Obama has weighed in.[1]

Arguably the main obstacle to automating ethical decisions is the lack of a formal specification of ground-truth *ethical principles*, which have been the subject of debate for centuries among ethicists and moral philosophers (Rawls 1971; Williams 1986). In their work on fairness in machine learning, Dwork et al. (2012) concede that, when ground-truth ethical principles are not available, we must use an "approximation as agreed upon by society." But how can society agree on the ground truth — or an approximation thereof — when even ethicists cannot?

We submit that decision making can, in fact, be automated, even in the absence of such ground-truth principles, by aggregating people's opinions on ethical dilemmas. This view is foreshadowed by recent position papers by Greene et al. (2016) and Conitzer et al. (2017), who suggest that the field of *computational social choice* (Brandt et al. 2016), which deals with algorithms for aggregating individual preferences towards collective decisions, may provide tools for ethical decision making. In particular, Conitzer et al. raise the possibility of "letting our *models* of multiple people's moral values *vote* over the relevant alternatives."

We take these ideas a step further by proposing and implementing a concrete approach for ethical decision making based on computational social choice, which, we believe, is quite practical. In addition to serving as a foundation for incorporating future ground-truth ethical and legal principles, it could even provide crucial preliminary guidance on some of the questions faced by ethicists. Our approach consists of four steps:

I *Data collection:* Ask human voters to compare pairs of alternatives (say a few dozen per voter). In the autonomous vehicle domain, an alternative is determined by a vector of features such as the number of victims and their gender, age, health — even species!

II *Learning:* Use the pairwise comparisons to learn a model of the preferences of each voter over all possible alternatives.

III *Summarization:* Combine the individual models into a single model, which approximately captures the collective preferences of all voters over all possible alternatives.

IV *Aggregation:* At runtime, when encountering an ethical dilemma involving a specific subset of alternatives, use the summary model to deduce the preferences of all voters over this particular subset, and apply a voting rule to aggregate these preferences into a collective decision. In the autonomous vehicle domain, the selected alternative is the outcome that society (as represented by the voters whose preferences were elicited in Step I) views as the least catastrophic among the grim options the vehicle currently faces. Note that this step is only applied when all other options have been exhausted, i.e., all technical ways of avoiding the dilemma in the first place have failed, and

[1] https://www.wired.com/2016/10/president-obama-mit-joi-ito-interview/

all legal constraints that may dictate what to do have also failed.

For Step I, we note that it is possible to collect an adequate dataset through, say, Amazon Mechanical Turk. But we actually perform this step on a much larger scale. Indeed, we use, for the first time, a unique dataset that consists of 18,254,285 pairwise comparisons between alternatives in the autonomous vehicle domain, obtained from 1,303,778 voters, through the website Moral Machine.[2]

Subsequent steps (namely Steps II, III, and IV) rely on having a *model* for preferences. There is a considerable line of work on distributions over rankings over a *finite* set of alternatives. A popular class of such models is the class of *random utility models*, which use random utilities for alternatives to generate rankings over the alternatives. We require a slightly more general notion, as we are interested in situations where the set of alternatives is infinite, and any finite subset of alternatives might be encountered (c.f. Caron and Teh 2012). For example, there are uncountably many scenarios an autonomous vehicle might face, because one can choose to model some features (such as the age of, say, a passenger) as continuous, but at runtime the vehicle will face a finite number of options. We refer to these generalized models as *permutation processes*.

In Section 3, we focus on developing a theory of aggregation of permutation processes, which is crucial for Step IV. Specifically, we assume that societal preferences are represented as a single permutation process. Given a (finite) subset of alternatives, the permutation process induces a distribution over rankings of these alternatives. In the spirit of *distributional rank aggregation* (Prasad, Pareek, and Ravikumar 2015), we view this distribution over rankings as an *anonymous preference profile*, where the probability of a ranking is the fraction of voters whose preferences are represented by that ranking. This means we can apply a voting rule in order to aggregate the preferences — but *which* voting rule should we apply? And how can we compute the outcome *efficiently*? These are some of the central questions in computational social choice, but we show that in our context, under rather weak assumptions on the voting rule and permutation process, they are both moot, in the sense that it is easy to identify alternatives chosen by any "reasonable" voting rule. In slightly more detail, we define the notion of *swap dominance* between alternatives in a preference profile, and show that if the permutation process satisfies a natural property with respect to swap dominance (standard permutation processes do), and the voting rule is *swap-dominance efficient* (all common voting rules are), then any alternative that swap dominates all other alternatives is an acceptable outcome.

Armed with these theoretical developments, our task can be reduced to: learning a permutation process for each voter (Step II); summarizing these individual processes into a single permutation process that satisfies the required swap-dominance property (Step III); and using any swap-dominance efficient voting rule, which is computationally efficient given the swap-dominance property (Step IV).

In Section 4, we present a concrete algorithm that instantiates our approach, for a specific permutation process, namely the Thurstone-Mosteller (TM) Process (Thurstone 1927; Mosteller 1951), and with a specific linear parametrization of its underlying utility process in terms of the alternative features. While these simple choices have been made to illustrate the framework, we note that, in principle, the framework can be instantiated with more general and complex permutation processes.

Finally, in Section 5, we implement and evaluate our algorithm. We first present simulation results from synthetic data that validate the accuracy of its learning and summarization components. More importantly, we implement our algorithm on the aforementioned Moral Machine dataset, and empirically evaluate the resultant system for choosing among alternatives in the autonomous vehicle domain. Taken together, these results suggest that our approach, and the algorithmic instantiation thereof, provide a computationally and statistically attractive method for ethical decision making.

2 Preliminaries

Let $\mathcal{X}$ denote a potentially infinite set of alternatives. Given a finite subset $A \subseteq \mathcal{X}$, we are interested in the set $\mathcal{S}_A$ of *rankings* over A. Such a ranking $\sigma \in \mathcal{S}_A$ can be interpreted as mapping alternatives to their positions, i.e., $\sigma(a)$ is the position of $a \in A$ (smaller is more preferred). Let $a \succ_\sigma b$ denote that a is preferred to b in σ, that is, $\sigma(a) < \sigma(b)$. For $\sigma \in \mathcal{S}_A$ and $B \subseteq A$, let $\sigma|_B$ denote the ranking σ restricted to B. And for a distribution P over $\mathcal{S}_A$ and $B \subseteq A$, define $P|_B$ in the natural way to be the restriction of P to B, i.e., for all $\sigma' \in \mathcal{S}_B$,

$$P|_B(\sigma') = \sum_{\sigma \in \mathcal{S}_A \colon \sigma|_B = \sigma'} P(\sigma).$$

A *permutation process* $\{\Pi(A) : A \subseteq \mathcal{X}, |A| \in \mathbb{N}\}$ is a collection of distributions over $\mathcal{S}_A$ for every finite subset of alternatives A. We say that a permutation process is *consistent* if $\Pi(A)|_B = \Pi(B)$ for any finite subsets of alternatives $B \subseteq A \subseteq \mathcal{X}$. In other words, for a consistent permutation process Π, the distribution induced by Π over rankings of the alternatives in B is nothing but the distribution obtained by marginalizing out the extra alternatives $A \setminus B$ from the distribution induced by Π over rankings of the alternatives in A. This definition of consistency is closely related to the Luce Choice Axiom (Luce 1959).

A simple adaptation of folklore results (Marden 1995) shows that any permutation process that is consistent has a natural interpretation in terms of utilities. Specifically (and somewhat informally, to avoid introducing notation that will not be used later), given any consistent permutation process Π over a set of alternatives $\mathcal{X}$ (such that $|\mathcal{X}| \leq \aleph_1$), there exists a stochastic process U (indexed by $\mathcal{X}$) such that for any $A = \{x_1, \ldots, x_m\} \subseteq \mathcal{X}$, the probability of drawing $\sigma \in \mathcal{S}_A$ from $\Pi(A)$ is equal to the probability that $\text{sort}(U_{x_1}, U_{x_2}, \cdots, U_{x_m}) = \sigma$, where (perhaps obviously) $\text{sort}(\cdot)$ sorts the utilities in non-increasing order. We can allow ties in utilities, as long as $\text{sort}(\cdot)$ is endowed with some tie-breaking scheme, e.g., ties are broken lexicographically,

[2] http://moralmachine.mit.edu

which we will assume in the sequel. We refer to the stochastic process corresponding to a consistent permutation process as its *utility process*, since it is semantically meaningful to obtain a permutation via sorting by utility.

As examples of natural permutation processes, we adapt the definitions of two well-known *random utility models*. The (relatively minor) difference is that random utility models define a distribution over rankings over a fixed, finite subset of alternatives, whereas permutation processes define a distribution for each finite subset of alternatives, given a potentially infinite space of alternatives.

- **Thurstone-Mosteller (TM) Process** (Thurstone 1927; Mosteller 1951). A Thurstone-Mosteller Process (adaptation of Thurstones Case V model) is a consistent permutation process, whose utility process U is a Gaussian process with independent utilities and identical variances. In more detail, given a finite set of alternatives $\{x_1, x_2, \cdots, x_m\}$, the utilities $(U_{x_1}, U_{x_2}, \cdots, U_{x_m})$ are independent, and $U_{x_i} \sim \mathcal{N}(\mu_{x_i}, \frac{1}{2})$, where μ_{x_i} denotes the mode utility of alternative x_i.
- **Plackett-Luce (PL) Process** (Plackett 1975; Luce 1959). A Plackett-Luce Process is a consistent permutation process with the following utility process U: Given a finite set of alternatives $\{x_1, x_2, \cdots, x_m\}$, the utilities $(U_{x_1}, U_{x_2}, \cdots, U_{x_m})$ are independent, and each U_{x_i} has a Gumbel distribution with identical scale, i.e. $U_{x_i} \sim \mathcal{G}(\mu_{x_i}, \gamma)$, where $\mathcal{G}$ denotes the Gumbel distribution, and μ_{x_i} denotes the mode utility of alternative x_i. We note that Caron and Teh (2012) consider a further Bayesian extension of the above PL process, with a Gamma process prior over the mode utility parameters.

3 Aggregation of Permutation Processes

In social choice theory, a *preference profile* is typically defined as a collection $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_N)$ of N rankings over a finite set of alternatives A, where σ_i represents the preferences of voter i. However, when the identity of voters does not play a role, we can instead talk about an *anonymous preference profile* $\pi \in [0,1]^{|A|!}$, where, for each $\sigma \in \mathcal{S}_A$, $\pi(\sigma) \in [0,1]$ is the *fraction* of voters whose preferences are represented by the ranking σ. Equivalently, it is the probability that a voter drawn uniformly at random from the population has the ranking σ.

How is this related to permutation processes? Given a permutation process Π and a finite subset $A \subseteq \mathcal{X}$, the distribution $\Pi(A)$ over rankings of A can be seen as an anonymous preference profile π, where for $\sigma \in \mathcal{S}_A$, $\pi(\sigma)$ is the probability of σ in $\Pi(A)$. As we shall see in Section 4, Step II (learning) gives us a permutation process for each voter, where $\pi(\sigma)$ represents our *confidence* that the preferences of the voter over A coincide with σ; and after Step III (summarization), we obtain a single permutation process that represents societal preferences.

Our focus in this section is the aggregation of anonymous preference profiles induced by a permutation process (Step IV), that is, the task of choosing the winning alternative(s). To this end, let us define an *anonymous social choice correspondence (SCC)* as a function f that maps any anonymous preference profile π over any finite and nonempty subset $A \subseteq \mathcal{X}$ to a nonempty subset of A. For example, under the ubiquitous *plurality* correspondence, the set of selected alternatives consists of alternatives with maximum first-place votes, i.e., $\arg\max_{a \in A} \sum_{\sigma \in \mathcal{S}_A :\, \sigma(a)=1} \pi(\sigma)$; and under the *Borda count* correspondence, denoting $|A| = m$, each vote awards $m - j$ points to the alternative ranked in position j, that is, the set of selected alternatives is $\arg\max_{a \in A} \sum_{j=1}^{m} (m-j) \sum_{\sigma \in \mathcal{S}_A :\, \sigma(a)=j} \pi(\sigma)$. We work with social choice *correspondences* instead of social choice *functions*, which return a single alternative in A, in order to smoothly handle ties.

3.1 Efficient Aggregation

Our main goal in this section is to address two related challenges. First, which (anonymous) social choice correspondence should we apply? There are many well-studied options, which satisfy different social choice axioms, and, in many cases, lead to completely different outcomes on the same preference profile. Second, how can we apply it in a computationally efficient way? This is not an easy task because, in general, we would need to explicitly construct the whole anonymous preference profile $\Pi(A)$, and then apply the SCC to it. The profile $\Pi(A)$ is of size $|A|!$, and hence this approach is intractable for a large $|A|$. Moreover, in some cases (such as the TM process), even computing the probability of a single ranking may be hard. The machinery we develop below allows us to completely circumvent these obstacles.

Since stating our general main result requires some setup, we first state a simpler instantiation of the result for the specific TM and PL permutation processes (we will directly use this instantiation in Section 4). Before doing so, we recall a few classic social choice axioms. We say that an anonymous SCC f is *monotonic* if the following conditions hold:

1. If $a \in f(\pi)$, and π' is obtained by pushing a upwards in the rankings, then $a \in f(\pi')$.
2. If $a \in f(\pi)$ and $b \notin f(\pi)$, and π' is obtained by pushing a upwards in the rankings, then $b \notin f(\pi')$.

In addition, an anonymous SCC is *neutral* if $f(\tau(\pi)) = \tau(f(\pi))$ for any anonymous preference profile π, and any permutation τ on the alternatives; that is, the SCC is symmetric with respect to the alternatives (in the same way that anonymity can be interpreted as symmetry with respect to voters).

Theorem 3.1. *Let Π be the TM or PL process, let $A \subseteq \mathcal{X}$ be a nonempty, finite subset of alternatives, and let $a \in \arg\max_{x \in A} \mu_x$. Moreover, let f be an anonymous SCC that is monotonic and neutral. Then $a \in f(\Pi(A))$.*

To understand the implications of the theorem, we first note that many of the common voting rules, including plurality, Borda count (and, in fact, all positional scoring rules), Copeland, maximin, and Bucklin (see, e.g., Brandt et al. 2016), are associated with anonymous, neutral, and monotonic SCCs. Specifically, all of these rules have a notion of score, and the SCC simply selects all the alternatives tied for the top score (typically there is only one).[3] The theorem then implies that all of these rules would agree that, given a subset of alternatives A, an alternative $a \in A$ with maximum mode utility is an acceptable winner, i.e., it is at least tied for the highest score, if it is not the unique winner. As we will see in Section 4, such an alternative is very easy to identify, which is why, in our view, Theorem 3.1 gives a satisfying solution to the challenges posed at the beginning of this subsection. We emphasize that this is merely an instantiation of Theorem 3.7, which provides our result for general permutation processes.

The rest of this subsection is devoted to building the conceptual framework, and stating the lemmas, required for the proof of Theorem 3.1, as well as the statement and proof of Theorem 3.7. We relegate all proofs to the full version of the paper (Noothigattu et al. 2017).

Starting off, let π denote an anonymous preference profile (or distribution over rankings) over alternatives A. We define the ranking σ^{ab} as the ranking σ with alternatives a and b swapped, i.e. $\sigma^{ab}(x) = \sigma(x)$ if $x \in A \setminus \{a, b\}$, $\sigma^{ab}(b) = \sigma(a)$, and $\sigma^{ab}(a) = \sigma(b)$.

Definition 3.2. We say that alternative $a \in A$ *swap-dominates* alternative $b \in A$ in anonymous preference profile π over A — denoted by $a \rhd_\pi b$ — if for every ranking $\sigma \in \mathcal{S}_A$ with $a \succ_\sigma b$ it holds that $\pi(\sigma) \geq \pi(\sigma^{ab})$.

In words, a swap-dominates b if every ranking that places a above b has at least as much weight as the ranking obtained by swapping the positions of a and b, and keeping everything else fixed. This is a very strong dominance relation, and, in particular, implies existing dominance notions such as *position dominance* (Caragiannis, Procaccia, and Shah 2016). Next we define a property of social choice correspondences, which intuitively requires that the correspondence adhere to swap dominance relations, if they exist in a given anonymous preference profile.

Definition 3.3. An anonymous SCC f is said to be *swap-dominance-efficient (SwD-efficient)* if for every anonymous preference profile π and any two alternatives a and b, if a swap-dominates b in π, then $b \in f(\pi)$ implies $a \in f(\pi)$.

Because swap-dominance is such a strong dominance relation, SwD-efficiency is a very weak requirement, which is intuitively satisfied by almost any "reasonable" voting rule. This intuition is formalized in the following lemma.

Lemma 3.4. *Any anonymous SCC that satisfies monotonicity and neutrality is SwD-efficient.*

So far, we have defined a property, SwD-efficiency, that any SCC might potentially satisfy. But why is this useful in the context of aggregating permutation processes? We answer this question in Theorem 3.7, but before stating it, we need to introduce the definition of a property that a *permutation process* might satisfy.

Definition 3.5. Alternative $a \in \mathcal{X}$ swap-dominates alternative $b \in \mathcal{X}$ in the permutation process Π — denoted by $a \rhd_\Pi b$ — if for every finite set of alternatives $A \subseteq \mathcal{X}$ such that $\{a, b\} \subseteq A$, a swap-dominates b in the anonymous preference profile $\Pi(A)$.

We recall that a *total preorder* is a binary relation that is transitive and total (and therefore reflexive).

Definition 3.6. A permutation process Π over $\mathcal{X}$ is said to be *SwD-compatible* if the binary relation $\rhd_\Pi$ is a total preorder on $\mathcal{X}$.

We are now ready to state our main theorem.

Theorem 3.7. *Let f be an SwD-efficient anonymous SCC, and let Π be an SwD-compatible permutation process. Then for any finite subset of alternatives A, there exists $a \in A$ such that $a \rhd_\Pi b$ for all $b \in A$. Moreover, $a \in f(\Pi(A))$.*

This theorem asserts that for any SwD-compatible permutation process, any SwD-efficient SCC (which, as noted above, include most natural SCCs, namely those that are monotonic and neutral), given any finite set of alternatives, will always select a very natural winner that swap-dominates other alternatives. A practical use of this theorem requires two things: to show that the permutation process is SwD-compatible, and that it is computationally tractable to select an alternative that swap-dominates other alternatives in a finite subset. The next few lemmas provide some general recipes for establishing these properties for general permutation processes, and, in particular, we show that they indeed hold under the TM and PL processes. First, we have the following definition.

Definition 3.8. Alternative $a \in \mathcal{X}$ *dominates* alternative $b \in \mathcal{X}$ in utility process U if for every finite subset of alternatives containing a and b, $\{a, b, x_3, \dots x_m\} \subseteq \mathcal{X}$, and every vector of utilities $(u_1, u_2, u_3 \dots u_m) \in \mathbb{R}^m$ with $u_1 \geq u_2$, it holds that

$$\begin{aligned} &p_{(U_a, U_b, U_{x_3}, \dots U_{x_m})}(u_1, u_2, u_3 \dots u_m) \\ &\quad \geq p_{(U_a, U_b, U_{x_3}, \dots U_{x_m})}(u_2, u_1, u_3 \dots u_m), \end{aligned} \tag{1}$$

where $p_{(U_a, U_b, U_{x_3}, \dots U_{x_m})}$ is the density function of the random vector $(U_a, U_b, U_{x_3}, \dots U_{x_m})$.

Building on this definition, Lemmas 3.9 and 3.10 directly imply that the TM and PL processes are SwD-compatible, and complete the proof of Theorem 3.1 (see the full version of the paper).

Lemma 3.9. *Let Π be a consistent permutation process, and let U be its corresponding utility process. If alternative a dominates alternative b in U, then a swap-dominates b in Π.*

Lemma 3.10. *Under the TM and PL processes, alternative a dominates alternative b in the corresponding utility process if and only if $\mu_a \geq \mu_b$.*

3.2 Stability

It turns out that the machinery developed for the proof of Theorem 3.1 can be leveraged to establish an additional desirable property.

[3] Readers who are experts in social choice have probably noted that there are no social choice *functions* that are both anonymous and neutral (Moulin 1983), intuitively because it is impossible to break ties in a neutral way. This is precisely why we work with social choice *correspondences*.

Definition 3.11. Given an anonymous SCC f, and a permutation process Π over $\mathcal{X}$, we say that the pair (Π, f) is *stable* if for any nonempty and finite subset of alternatives $A \subseteq \mathcal{X}$, and any nonempty subset $B \subseteq A$, $f(\Pi(A)) \cap B = f(\Pi(B))$ whenever $f(\Pi(A)) \cap B \neq \phi$.

Intuitively, stability means that applying f under the assumption that the set of alternatives is A, and then reducing to its subset B, is the same as directly reducing to B and then applying f. This notion is related to classic axioms studied by Sen (1971), specifically his *expansion* and *contraction* properties. In our setting, stability seems especially desirable, as our algorithm would potentially face decisions over many different subsets of alternatives, and the absence of stability may lead to glaringly inconsistent choices.

Theorem 3.12. *Let* Π *be the TM or PL process, and let* f *be the Borda count or Copeland SCC. Then the pair* (Π, f) *is stable.*

The definition of the Copeland SCC, and the proof of the theorem, are relegated to the full version of the paper (Noothigattu et al. 2017). Among other things, the proof requires a stronger notion of SwD-efficiency, which, as we show, is satisfied by Borda count and Copeland, and potentially by other appealing SCCs.

4 Instantiation of Our Approach

In this section, we instantiate our approach for ethical decision making, as outlined in Section 1. In order to present a concrete algorithm, we consider a specific permutation process, namely the TM process with a linear parameterization of the utility process parameters as a function of the alternative features.

Let the set of alternatives be given by $\mathcal{X} \subseteq \mathbb{R}^d$, i.e. each alternative is represented by a vector of d features. Furthermore, let N denote the total number of voters. Assume for now that the data-collection step (Step I) is complete, i.e., we have some pairwise comparisons for each voter; we will revisit this step in Section 5.

Step II: Learning. For each voter, we learn a TM process using his pairwise comparisons to represent his preferences. We assume that the mode utility of an alternative x depends linearly on its features, i.e., $\mu_x = \boldsymbol{\beta}^\intercal x$. Note that we do not need an intercept term, since we care only about the relative ordering of utilities. Also note that the parameter $\boldsymbol{\beta} \in \mathbb{R}^d$ completely describes the TM process, and hence the parameters $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots \boldsymbol{\beta}_N$ completely describe the models of all voters.

Next we provide a computationally efficient method for learning the parameter $\boldsymbol{\beta}$ for a particular voter. Let $(X_1, Z_1), (X_2, Z_2), \cdots, (X_n, Z_n)$ denote the pairwise comparison data of the voter. Specifically, the ordered pair (X_j, Z_j) denotes the j^{th} pair of alternatives compared by the voter, and the fact that the voter chose X_j over Z_j. We use maximum likelihood estimation to estimate $\boldsymbol{\beta}$. The log-likelihood function is

$$\mathcal{L}(\boldsymbol{\beta}) = \log \left[\prod_{j=1}^{n} P(X_j \succ Z_j; \boldsymbol{\beta}) \right]$$

$$= \sum_{j=1}^{n} \log P(U_{X_j} > U_{Z_j}; \boldsymbol{\beta})$$

$$= \sum_{j=1}^{n} \log \Phi \left(\boldsymbol{\beta}^\intercal (X_j - Z_j) \right),$$

where Φ is the cumulative distribution function of the standard normal distribution, and the last transition holds because $U_x \sim \mathcal{N}(\boldsymbol{\beta}^\intercal x, \frac{1}{2})$. Note that the standard normal CDF Φ is a log-concave function. This makes the log-likelihood concave in $\boldsymbol{\beta}$, hence we can maximize it efficiently.

Step III: Summarization. After completing Step II, we have N TM processes represented by the parameters $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots \boldsymbol{\beta}_N$. In Step III, we bundle these individual models into a single permutation process $\hat{\Pi}$, which, in the current instantiation, is also a TM process with parameter $\hat{\boldsymbol{\beta}}$ (see Section 6 for a discussion of this point). We perform this step because we must be able to make decisions *fast*, in Step IV. For example, in the autonomous vehicle domain, the AI would only have a split second to make a decision in case of emergency; aggregating information from millions of voters *in real time* will not do. By contrast, Step III is performed offline, and provides the basis for fast aggregation.

Let $\Pi^{\boldsymbol{\beta}}$ denote the TM process with parameter $\boldsymbol{\beta}$. Given a finite subset of alternatives $A \subseteq \mathcal{X}$, the anonymous preference profile generated by the model of voter i is given by $\Pi^{\boldsymbol{\beta}_i}(A)$. Ideally, we would like the summary model to be such that the profile generated by it, $\hat{\Pi}(A)$, is as close as possible to $\Pi^*(A) = \frac{1}{N}\sum_{i=1}^{N} \Pi^{\boldsymbol{\beta}_i}(A)$, the mean profile obtained by giving equal importance to each voter. However, there does not appear to be a straightforward method to compute the "best" $\hat{\boldsymbol{\beta}}$, since the profiles generated by the TM processes do not have an explicit form. Hence, we use utilities as a proxy for the quality of $\hat{\boldsymbol{\beta}}$. Specifically, we find $\hat{\boldsymbol{\beta}}$ such that the summary model induces utilities that are as close as possible to the mean of the utilities induced by the per-voter models, i.e., we want $U_x^{\hat{\boldsymbol{\beta}}}$ to be as close as possible (in terms of KL divergence) to $\frac{1}{N}\sum_{i=1}^{N} U_x^{\boldsymbol{\beta}_\mathbf{i}}$ for each $x \in \mathcal{X}$, where $U_x^{\boldsymbol{\beta}}$ denotes the utility of x under TM process with parameter $\boldsymbol{\beta}$. This is achieved by taking $\hat{\boldsymbol{\beta}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\beta}_i$, as shown by the following proposition, whose proof appears in the full version of the paper (Noothigattu et al. 2017).

Proposition 4.1. *The vector* $\boldsymbol{\beta} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\beta}_i$ *minimizes* $KL\left(\frac{1}{N}\sum_{i=1}^{N} U_x^{\boldsymbol{\beta}_i} \middle\| U_x^{\boldsymbol{\beta}}\right)$ *for any* $x \in \mathcal{X}$.

Step IV: Aggregation. As a result of Step III, we have exactly one (summary) TM process $\hat{\Pi}$ (with parameter $\hat{\boldsymbol{\beta}} = \bar{\boldsymbol{\beta}}$) to work with at runtime. Given a finite set of alternatives $A = \{x_1, x_2, \cdots, x_m\}$, we must aggregate the preferences represented by the anonymous preference profile $\hat{\Pi}(A)$. This is where the machinery of Section 3 comes in: We simply need to select an alternative that has maximum mode utility among $\hat{\boldsymbol{\beta}}^\intercal x_1, \hat{\boldsymbol{\beta}}^\intercal x_2, \cdots, \hat{\boldsymbol{\beta}}^\intercal x_m$. Such an alternative would be selected by any anonymous SCC that is monotonic and neutral, when applied to $\hat{\Pi}(A)$, as shown by Theorem 3.1. Moreover, this aggregation method is equivalent to applying the Borda count or Copeland SCCs. Hence, we also have the desired stability property, as shown by Theorem 3.12.

5 Implementation and Evaluation

In this section, we implement the algorithm presented in Section 4, and empirically evaluate it. We start with an implementation on synthetic data, which allows us to effectively validate both Steps II and III of our approach. We then describe the Moral Machine dataset mentioned in Section 1, present the implementation of our algorithm on this dataset, and evaluate the resultant system for ethical decision making in the autonomous vehicle domain (focusing on Step III).

5.1 Synthetic Data

Setup. We represent the preferences of each voter using a TM process. Let $\boldsymbol{\beta}_i$ denote the true parameter corresponding to the model of voter i. We sample $\boldsymbol{\beta}_i$ from $\mathcal{N}(\mathbf{m}, I_d)$ (independently for each voter i), where each mean m_j is sampled independently from the uniform distribution $\mathcal{U}(-1, 1)$, and the number of features is $d = 10$.

In each instance (defined by a subset of alternatives A with $|A| = 5$), the desired winner is given by the application of Borda count to the mean of the profiles of the voters. In more detail, we compute the anonymous preference profile of each voter $\Pi^{\boldsymbol{\beta}_i}(A)$, and then take a mean across all the voters to obtain the desired profile $\frac{1}{N}\sum_{i=1}^{N} \Pi^{\boldsymbol{\beta}_i}(A)$. We then apply Borda count to this profile to obtain the winner. Note that, since we are dealing with TM processes, we cannot explicitly construct $\Pi^{\boldsymbol{\beta}_i}(A)$; we therefore estimate it by sampling rankings according to the TM process of voter i.

Evaluation of Step II (Learning). In practice, the algorithm does not have access to the true parameter $\boldsymbol{\beta}_i$ of voter i, but only to pairwise comparisons, from which we learn the parameters. Thus we compare the computation of the winner (following the approach described above) using the true parameters, and using the learned parameters as in Step II. We report the accuracy as the fraction of instances, out of 100 test instances, in which the two outcomes match.

To generate each pairwise comparison of voter i, for each of $N = 20$ voters, we first sample two alternatives x_1 and x_2 independently from $\mathcal{N}(\mathbf{0}, I_d)$. Then, we sample their utilities U_{x_1} and U_{x_2} from $\mathcal{N}(\boldsymbol{\beta}_i^\intercal x_1, \frac{1}{2})$ and $\mathcal{N}(\boldsymbol{\beta}_i^\intercal x_2, \frac{1}{2})$, respectively. Of course, the voter prefers the alternative with higher sampled utility. Once we have the comparisons, we learn the parameter $\boldsymbol{\beta}_i$ by computing the MLE (as explained in Step II of Section 4). In our results, we vary the number of pairwise comparisons per voter and compute the accuracy to obtain the learning curve shown in Figure 1. Each datapoint in the graph is averaged over 50 runs. Observe that the accuracy quickly increases as the number of pairwise comparisons increases, and with just 30 pairwise comparisons we achieve an accuracy of 84.3%. With 100 pairwise comparisons, the accuracy is 92.4%.

Evaluation of Step III (Summarization). To evaluate Step III, we assume that we have access to the true parameters $\boldsymbol{\beta}_i$, and wish to determine the accuracy loss incurred in the summarization step, where we summarize the individual TM models into a single TM model. As described in Section 4, we compute $\bar{\boldsymbol{\beta}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\beta}_i$, and, given a subset A (which again has cardinality 5), we aggregate using Step IV, since we now have just one TM process. For each instance, we contrast our computed winner with the desired winner as computed previously. We vary the number of voters and compute the accuracy to obtain Figure 2. The accuracies are averaged over 50 runs. Observe that the accuracy increases to 93.9% as the number of voters increases. In practice we expect to have access to thousands, even millions, of votes (see Section 5.2). We conclude that, surprisingly, the expected loss in accuracy due to summarization is quite small.

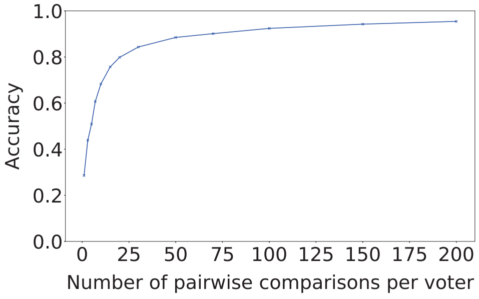

Figure 1: Accuracy of Step II (synthetic data)

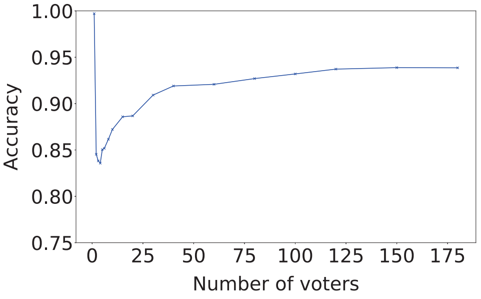

Figure 2: Accuracy of Step III (synthetic data)

Robustness. Our results are robust to the choice of parameters, as we demonstrate in the full version of the paper (Noothigattu et al. 2017).

5.2 Moral Machine Data

Moral Machine is a platform for gathering data on human perception of the moral acceptability of decisions made by autonomous vehicles faced with choosing which humans to harm and which to save. The main interface of Moral Machine is the Judge mode. This interface generates sessions of random moral dilemmas. In each session, a user is faced with 13 instances. Each instance features an autonomous vehicle with a brake failure, facing a moral dilemma with two possible alternatives, that is, each instance is a pairwise comparison. Each of the two alternatives corresponds to sacrificing the lives of one group of characters to spare those of another group of characters. Figure 3 shows an example of

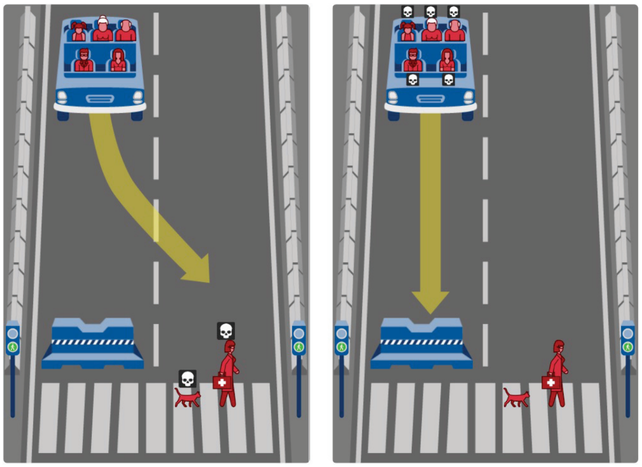

Figure 3: *Moral Machine* — Judge interface. This particular choice is between a group of pedestrians that includes a female doctor and a cat crossing on a green light, and a group of passengers including a woman, a male executive, an elderly man, an elderly woman, and a girl.

such an instance. Respondents choose the outcome that they prefer the autonomous vehicle to make.

Each alternative is characterized by 22 features: relation to the autonomous vehicle (passengers or pedestrians), legality (no legality, explicitly legal crossing, or explicitly illegal crossing), and counts of 20 character types, including ones like man, woman, pregnant woman, male athlete, female doctor, dog, etc. When sampling from the 20 characters, some instances are generated to have an easy-to-interpret tradeoff with respect to some dimension, such as gender (males on one side vs. females on the other), age (elderly vs. young), fitness (large vs. fit), etc., while other instances have groups consisting of completely randomized characters being sacrificed in either alternative. Alternatives with all possible combinations of these features are considered, except for the legality feature in cases when passengers are sacrificed. In addition, each alternative has a derived feature, "number of characters," which is simply the sum of counts of the 20 character types (making $d = 23$).

As mentioned in Section 1, the Moral Machine dataset consists of preference data from 1,303,778 voters, amounting to a total of 18,254,285 pairwise comparisons. We used this dataset to learn the $\boldsymbol{\beta}$ parameters of all 1.3 million voters (Step II, as given in Section 4). Next, we took the mean of all of these $\boldsymbol{\beta}$ vectors to obtain $\hat{\boldsymbol{\beta}}$ (Step III). This gave us an implemented system, which can be used to make real-time choices between any finite subset of alternatives.

Importantly, the methodology we used, in Section 5.1, to evaluate Step II on the synthetic data cannot be applied to the Moral Machine data, because we do not know which alternative would be selected by aggregating the preferences of the actual 1.3 million voters over a subset of alternatives. However, we can apply a methodology similar to that of Section 5.1 in order to evaluate Step III. Specifically, as in Section 5.1, we wish to compare the winner obtained using the summarized model, with the winner obtained by applying Borda count to the mean of the anonymous preference profiles of the voters.

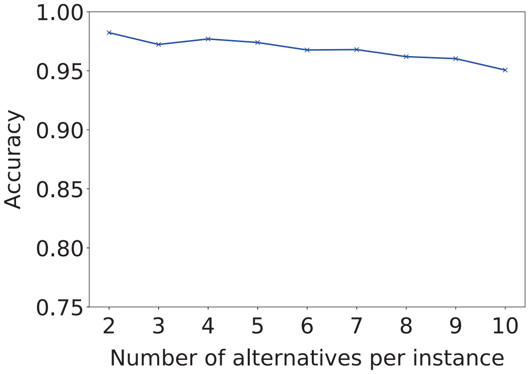

Figure 4: Accuracy of Step III (Moral Machine data)

An obstacle is that now we have a total of 1.3 million voters, and hence it would take an extremely long time to calculate the anonymous preference profile of each voter and take their mean (this was the motivation for having Step III in the first place). So, instead, we estimate the mean profile by sampling rankings, i.e., we sample a voter i uniformly at random, and then sample a ranking from the TM process of voter i; such a sampled ranking is an i.i.d. sample from the mean anonymous profile. Then, we apply Borda count as before to obtain the desired winner (note that this approach is still too expensive to use in real time). The winner according to the summarized model is computed exactly as before, and is just as efficient even with 1.3 million voters.

Using this methodology, we computed accuracy on 3000 test instances, i.e., the fraction of instances in which the two winners match. Figure 4 shows the results as the number of alternatives per instance is increased from 2 to 10. Observe that the accuracy is as high as 98.2% at 2 alternatives per instance, and gracefully degrades to 95.1% at 10.

6 Discussion

The design of intelligent machines that can make ethical decisions is, arguably, one of the hardest challenges in AI. We do believe that our approach takes a significant step towards addressing this challenge. In particular, the implementation of our algorithm on the Moral Machine dataset has yielded a system which, arguably, can make *credible* decisions on ethical dilemmas in the autonomous vehicle domain (when all other options have failed). But this paper is clearly not the end-all solution.

Most important is the (primarily conceptual) challenge of extending our framework to incorporate ethical or legal principles — at least for simpler settings where they might be easier to specify. The significant advantage of having our approach in place is that these principles do not need to always lead to a decision, as we can fall back on the societal choice. This allows for a modular design where principles are incorporated over time, without compromising the ability to make a decision in every situation.

In addition, as mentioned in Section 4, we have made some specific choices to instantiate our approach. We discuss two of the most consequential choices. First, we assume that the mode utilities have a linear structure. This means that, under the TM model, the estimation of the maximum likelihood parameters is a convex program (see Section 4), hence we can learn the preferences of millions of voters, as in the Moral Machine dataset. Moreover, a straightforward summarization method works well. However, dealing with a richer representation for utilities would require new methods for both learning and summarization (Steps II and III).

Second, the instantiation given in Section 4 summarizes the N individual TM models as a single TM model. While the empirical results of Section 5 suggest that this method is quite accurate, even higher accuracy can potentially be achieved by summarizing the N models as a *mixture* of K models, for a relatively small K. This leads to two technical challenges: What is a good algorithm for generating this mixture of, say, TM models? And, since the framework of Section 3 would not apply, how should such a mixture be aggregated — does the (apparently mild) increase in accuracy come at great cost to computational efficiency?

Finally, it is worth noting that, in parallel work, Freedman et al. (2018) introduce a related approach, and apply it to the problem of prioritizing patients in kidney exchange. Specifically, they collect preferences from 289 workers on Amazon Mechanical Turk, and use them to learn societal weights for all types of patients. These weights are then used to break ties among multiple maximum-cardinality matchings between patients and donors. In contrast to our approach, aggregation essentially happens in the learning phase, i.e., a voting rule is never explicitly applied. Nonetheless, their work serves as another compelling proof of concept (in a different domain), providing additional evidence that ethical decisions can be automated through computational social choice and machine learning.

Acknowledgments

This work was partially supported by NSF grants IIS-1350598, IIS-1714140, IIS-1149803, CCF-1525932, and CCF-1733556; by ONR grants N00014-16-1-3075 and N00014-17-1-2428; by two Sloan Research Fellowships; and by the Ethics & Governance of AI Fund.

References

Bonnefon, J.-F.; Shariff, A.; and Rahwan, I. 2016. The social dilemma of autonomous vehicles. *Science* 352(6293):1573–1576.

Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D., eds. 2016. *Handbook of Computational Social Choice*. Cambridge University Press.

Caragiannis, I.; Procaccia, A. D.; and Shah, N. 2016. When do noisy votes reveal the truth? *ACM Transactions on Economics and Computation* 4(3): article 15.

Caron, F., and Teh, Y. W. 2012. Bayesian nonparametric models for ranked data. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*, 1529–1537.

Conitzer, V.; Sinnott-Armstrong, W.; Schaich Borg, J.; Deng, Y.; and Kramer, M. 2017. Moral decision making frameworks for artificial intelligence. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, 4831–4835.

Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; and Zemel, R. S. 2012. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS)*, 214–226.

Freedman, R.; Schaich Borg, J.; Sinnott-Armstrong, W.; Dickerson, J. P.; and Conitzer, V. 2018. Adapting a kidney exchange algorithm to align with human values. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*. Forthcoming.

Greene, J.; Rossi, F.; Tasioulas, J.; Venable, K. B.; and Williams, B. 2016. Embedding ethical principles in collective decision support systems. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, 4147–4151.

Jarvis Thomson, J. 1985. The trolley problem. *The Yale Law Journal* 94(6):1395–1415.

Luce, R. D. 1959. *Individual Choice Behavior: A Theoretical Analysis*. Wiley.

Marden, J. I. 1995. *Analysing and Modeling Rank Data*. Chapman & Hall.

Mosteller, F. 1951. Remarks on the method of paired comparisons: I. the least squares solution assuming equal standard deviations and equal correlations. *Psychometrika* 16(1):3–9.

Moulin, H. 1983. *The Strategy of Social Choice*, volume 18 of *Advanced Textbooks in Economics*. North-Holland.

Noothigattu, R.; Gaikwad, S. S.; Awad, E.; Dsouza, S.; Rahwan, I.; Ravikumar, P.; and Procaccia, A. D. 2017. A voting-based system for ethical decision making. arXiv:1709.06692 [cs.AI].

Plackett, R. 1975. The analysis of permutations. *Applied Statistics* 24:193–202.

Prasad, A.; Pareek, H. H.; and Ravikumar, P. 2015. Distributional rank aggregation, and an axiomatic analysis. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2104–2112.

Rawls, J. 1971. *A Theory of Justice*. Harvard University Press.

Sen, A. K. 1971. Choice functions and revealed preference. *Review of Economic Studies* 38(3):307–317.

Thurstone, L. L. 1927. A law of comparative judgement. *Psychological Review* 34:273–286.

Wallach, W., and Allen, C. 2008. *Moral Machines: Teaching Robots Right from Wrong*. Oxford University Press.

Williams, B. 1986. *Ethics and the Limits of Philosophy*. Harvard University Press.