Ethically Compliant Sequential Decision Making

Justin Svegliato • Samer Nashed • Shlomo Zilberstein

UMass Amherst
Motivation

• **Value alignment** is a notorious challenge for AI systems.
• **Ethical theories** offer guiding principles for scalable application of human values to machine decision making.
• However, despite extensive study, they are still **hard** to implement.
• Why? Developers must often **implicitly balance safe, ethical behavior with efficient behavior** during the design process.
Ethically Compliant Autonomous Systems

Task Completion is modeled by

Decision-Making Model

(maximize)

Optimal Amoral Policy
Ethically Compliant Autonomous Systems

Task Completion

- Task
  - is modeled by Decision-Making Model
    - maximize

Ethical Compliance

- Ethical Framework
  - is modeled by Ethical Context, Moral Principle
    - within, subject to

- Optimal Moral Policy
Markov Decision Processes

\[ \langle S, A, T, R, d \rangle \]

- states: \( S \)
- actions: \( A \)
- transition function: \( T : S \times A \times S \to [0, 1] \)
- reward function: \( R : S \times A \times S \to \mathbb{R} \)
- initial state function: \( d : S \to [0, 1] \)
Markov Decision Processes

policy \( \pi : S \rightarrow A \)

value function \( V^\pi : S \rightarrow \mathbb{R} \)

optimal policy \( \pi^* : S \rightarrow A \)

\[
\text{maximize } V^\pi_{\pi \in \Pi}
\]

The optimization problem can be solved using the primal form or the dual form of a linear program.
Task Completion

Task Completion

Task

is modeled by

Decision-Making Model

decision-making model

\[ D = \langle S, A, T, R, d \rangle \]

This represents the \textit{information} needed to complete the \textit{task}.
Ethical Compliance

ethical context

\[ \mathcal{E} = \langle \cdots \rangle \]

This represents the information needed to follow the ethical framework.

This information **may not be relevant to task completion!**
Ethical Compliance

moral principle

\[ \rho : \Pi \rightarrow \mathbb{B} \]

This evaluates whether the policy of the decision-making model within the ethical context is ethically compliant.
Constrained Optimization

\[
\text{maximize } V^\pi_{\pi \in \Pi}
\]

subject to \(\rho(\pi; D, E)\)

Diagram:
- \(\Pi_\rho\)
- \(\Pi_{-\rho}\)
- \(\pi^*_\rho\)
- \(\pi^*_\)
Example: Prima Facie Duties

The morality of an action is based on whether that action fulfills fundamental moral duties that can contradict each other.

1. Fidelity
2. Reparation
3. Gratitude
4. Non-Injury
5. Harm-Prevention

...
Example: Prima Facie Duties

The morality of an action is based on whether that action fulfills fundamental moral duties that can contradict each other.

ethical context

\[ \mathcal{E}_\Delta = \langle \Delta, \phi, \tau \rangle \]

duties \( \Delta \)

penalty function \( \phi : \Delta \times S \rightarrow \mathbb{R}^+ \)

tolerance \( \tau \in \mathbb{R}^+ \)
Example: Prima Facie Duties

The morality of an action is based on whether that action fulfills fundamental moral duties that can contradict each other.

\[
\rho_{\Delta}(\pi) = \sum_{s \in S} d(s)J^{\pi}(s) \leq \tau
\]

moral principle

\[
J^{\pi}(s) = \sum_{s' \in S} T(s, \pi(s), s')[\sum_{\delta \in \Delta_{s'}} \phi(\delta, s') + J^{\pi}(s')]
\]

expected cumulative penalty
User Study

Experts had to complete two tasks in a random order to satisfy a list of moral requirements.

**PFD Implementation Task:** Generate moral behavior by *implementing* the ethical context of prima facie duties.

**MDP Modification Task:** Generate moral behavior by *modifying* the reward function of the MDP.
User Study Results

Lower Left
Conservative Policies • Slower • Moral

Upper Right
Aggressive Policies • Faster • Immoral

Center Line
Goldilocks Policies
Morality.js is an **open-source** JavaScript library for building **autonomous systems** that comply with **ethical theories**

**Customizable Playground**

**Simple Code**

```javascript
import morality from 'morality';
import agents from 'morality/agents';
import ethics from 'morality/ethics';


const ethics = new ethics.DivineCommandTheory(0, 4, 10);

const solution = morality.solve(agent, ethics);
```