Norm Conflict Resolution in Stochastic Domains

Daniel Kasenberg Matthias Scheutz

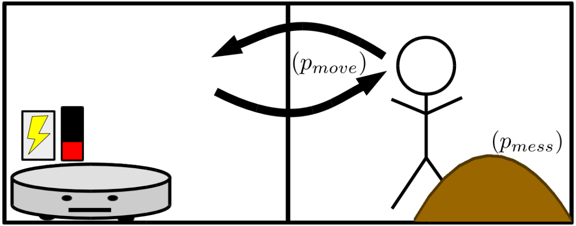
February 6, 2018

Human-Robot Interaction Laboratory (HRILab)
Tufts University

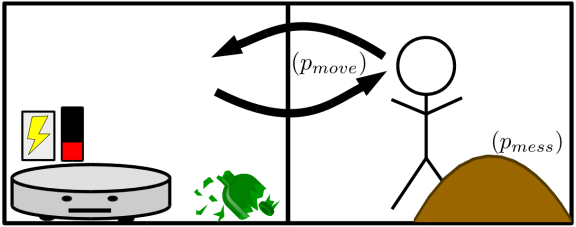
Introduction

- Artificial agents will need to be aware of human *moral and social norms*...
- ...and use them in decision-making
- Complication: norms *may conflict*

VacuumWorld



VacuumWorld: Norm Conflict



Logic- and reward-based systems

- Logic-based approaches to normative reasoning
  - Often use deontic logic for sophisticated normative reasoning
  - Inconsistent principles $\rightarrow$ normative explosion (e.g., everything obligated)
  - Usually not well-suited to stochastic environments
- Reward-based approaches to normative behavior
  - Encode norms implicitly, using reward functions
  - Difficult to interpret, explain, generalize to new domains

Our contribution

- A *hybrid approach*, employing ideas from logic- and reward-based approaches
- Represent norms in linear temporal logic (LTL)
- Agents in Markov Decision Process
- Deal with conflicts by minimizing a notion of ‘violation cost’

Linear temporal logic (LTL)

A propositional logic encoding time

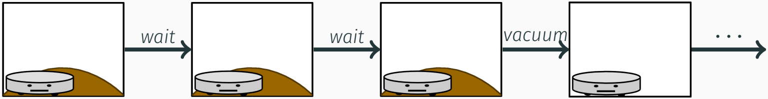
$$\phi ::= p \mid \neg\phi_1 \mid \phi_1 \vee \phi_2 \mid \phi_1 \wedge \phi_2 \mid \phi_1 \rightarrow \phi_2 \mid$$
$$\mathbf{X}\phi_1 \mid \mathbf{G}\phi_1 \mid \mathbf{F}\phi_1 \mid \phi_1 \mathbf{U} \phi_2$$

where ϕ_1, ϕ_2 are LTL statements; p a proposition from some set Π.

- $\mathbf{X}\phi_1$: "in the next time step, ϕ_1"
- $\mathbf{G}\phi_1$: "in all present and future time steps, ϕ_1"
- $\mathbf{F}\phi_1$: "in some present or future time step, ϕ_1"
- $\phi_1 \mathbf{U} \phi_2$: "ϕ_1 will be true until ϕ_2 becomes true"

$\mathbf{G}\mathit{roomClean}, \mathbf{G}\neg\mathit{robotDamaged}, \mathbf{G}\neg\mathit{humanInjured}$

Relating MDPs to LTL



$$\tau = \quad s_0, \quad a_0, \quad s_1, \quad a_1, \quad s_2, \quad a_2, \quad s_3,$$

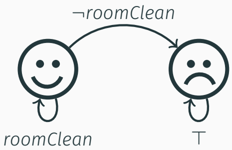
s_0	s_1	s_2	s_3
$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$
¬*roomClean*	¬*roomClean*	¬*roomClean*	*roomClean*
¬*batteryDead*	¬*batteryDead*	¬*batteryDead*	¬*batteryDead*

- Augment the MDP with a set Π of atomic propositions (e.g. *roomClean*, *batteryDead*)
- $\mathcal{L}(s)$: which propositions true in state s (*valuation* of s)
- LTL formulas are evaluated over an infinite sequence of *valuations* $\sigma_1, \sigma_2, \cdots$; that is, $\sigma_1, \sigma_2, \cdots \vDash \phi$
- We say that $\tau \vDash \phi$ iff $\mathcal{L}(s_0), \mathcal{L}(s_1), \cdots \vDash \phi$

Deterministic Rabin Automata (DRAs)

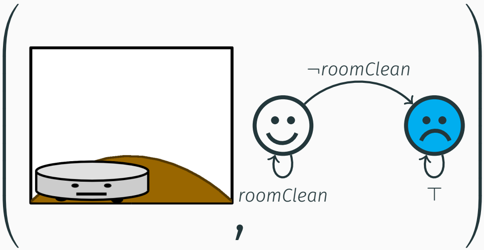
- Each LTL statement ϕ has a corresponding *Deterministic Rabin Automaton* $\mathcal{D}(\phi)$
  - A finite state machine over infinite words
  - Accepts if and only if the LTL statement is satisfied
- **Contains all information about agent's history relevant to the statement ϕ**

G*roomClean*



Product MDP

- Can construct a *product MDP* where each “product state” corresponds to a state of the original MDP, plus a DRA state
- The optimal course of action is *stationary* in the product space
    - i.e. depends only on the agent’s current product state.



LTL Planning (Ding et al. 2011)

- **Goal**: Maximize the probability of specifying a given LTL statement ϕ
- Compute the product MDP $\mathcal{M}^\times$ from the Deterministic Rabin Automaton $\mathcal{D}(\phi)$
- Determine a set of “good states” from which the agent is guaranteed to satisfy the LTL statement by following a certain policy
- Now a reachability problem: maximize the probability of reaching a set of good states in the product MDP
    - Can be solved by linear programming
- Result: optimal policy, stationary in the product space (but generally not in original MDP)

Planning and norm conflicts

- **Goal**: Satisfy a **set** of LTL statements $\phi_1, \cdots, \phi_n$ “as well as possible”.
- Could use the method of (Ding et al. 2011) with $\bigwedge_{i=1}^{n} \phi_i$
- But **probability of satisfying all norms might be zero**
- We say that a **norm conflict** has occurred when the probability of an agent satisfying all of its norms $\phi_1, \cdots, \phi_n$ is **zero**
- Max probability algorithms don’t help decision making in norm conflicts

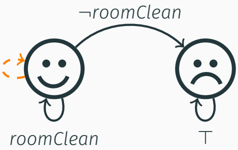
Violation cost

- To resolve norm conflicts, define some notion of ‘badness’ of norm violations and minimize it
- **Idea:** allow agent to temporarily “suspend” a norm (for a time step), but pay a cost for doing so
- Give each norm a weight w
- Agent’s goal is to minimize the expected weighted sum of costs

Conflict resolution DRA

- Can measure violation cost for a norm by adding self-loops in the DRA (we call the modified DRA a **conflict resolution DRA**)
- The agent takes the self-loop instead of entering 'bad' DRA states, but incurs a cost

$\mathbf{G}\,roomClean$



Minimizing expected violation cost

- At each time step, after seeing a new state s', decide which norms should be suspended.
- Let $\tilde{a}_i = 1$ iff ϕ_i is suspended
- The DRA for ϕ_i will transition from state q_i to state

$$q'_i = \begin{cases} q_i & \text{if } \tilde{a}_i = 1 \\ \delta_i(q_i, \mathcal{L}(s')) & \text{otherwise} \end{cases}$$

  where δ_i is the transition function of $\mathcal{D}(\phi_i)$
- The optimal total violation cost from a product state $(s, q_1, \cdots, q_n)$ satisfies the following equation:

$$\begin{aligned} Viol_{\mathcal{N}}((s, q_1, \cdots, q_n)) = \min_{a \in A} \sum_{s' \in S} T(s, a, s') \min_{\tilde{a} \in \{0,1\}^n} \sum_{i=1}^{n} \Big(& w_i \tilde{a}_i \\ & + \gamma Viol_{\mathcal{N}}((s', q'_1, \cdots, q'_n)) \Big) \end{aligned}$$

Minimizing expected violation cost

We can use *value iteration* to compute the optimal expected violation cost starting from each state:

$$Viol_{\mathcal{N}}^{(k+1)}((s, q_1, \cdots, q_n)) \leftarrow \min_{a \in A} \sum_{s' \in S} T(s, a, s') min_{\tilde{a} \in \{0,1\}^n} \sum_{i=1}^{n} \Big(w_i \tilde{a}_i + \gamma Viol_{\mathcal{N}}^{(k)}((s', q'_1, \cdots, q'_n)) \Big)$$

- To find the best action(s) from $(s, q_1, \cdots, q_n)$: take the $\arg\min_{a \in A}$
- This gives a *product-space policy* $\pi^* : S^{\otimes} \to A$

Norm conflict resolution algorithm

Given norm system $\mathcal{N} = \{(w_1, \phi_1), \cdots, (w_n, \phi_n)\}$, MDP $\mathcal{M} = \langle S, A, T, R, s_0, \gamma \rangle$:

Before acting ($t = 0$):

Norm conflict resolution algorithm

Given norm system $\mathcal{N} = \{(w_1, \phi_1), \cdots, (w_n, \phi_n)\}$, MDP $\mathcal{M} = \langle S, A, T, R, s_0, \gamma \rangle$:

Before acting ($t = 0$):

- Compute conflict resolution DRAs $\mathcal{D}(\phi_i)$ and product MDP $\mathcal{M}^{\otimes}$

Norm conflict resolution algorithm

Given norm system $\mathcal{N} = \{(w_1, \phi_1), \cdots, (w_n, \phi_n)\}$, MDP $\mathcal{M} = \langle S, A, T, R, s_0, \gamma \rangle$:

Before acting ($t = 0$):

- Compute conflict resolution DRAs $\mathcal{D}(\phi_i)$ and product MDP $\mathcal{M}^{\otimes}$
- Graph-theoretic stuff (ensure agent stays in good DRA states)

Norm conflict resolution algorithm

Given norm system $\mathcal{N} = \{(w_1, \phi_1), \cdots, (w_n, \phi_n)\}$, MDP $\mathcal{M} = \langle S, A, T, R, s_0, \gamma \rangle$:

Before acting ($t = 0$):

- Compute conflict resolution DRAs $\mathcal{D}(\phi_i)$ and product MDP $\mathcal{M}^{\otimes}$
- Graph-theoretic stuff (ensure agent stays in good DRA states)
- Compute $Viol_{\mathcal{N}}$ using value iteration

Norm conflict resolution algorithm

Given norm system $\mathcal{N} = \{(w_1, \phi_1), \cdots, (w_n, \phi_n)\}$, MDP $\mathcal{M} = \langle S, A, T, R, s_0, \gamma \rangle$:

Before acting ($t = 0$):

- Compute conflict resolution DRAs $\mathcal{D}(\phi_i)$ and product MDP $\mathcal{M}^{\otimes}$
- Graph-theoretic stuff (ensure agent stays in good DRA states)
- Compute $Viol_{\mathcal{N}}$ using value iteration
- Use $Viol_{\mathcal{N}}$ to determine optimal product-space policy π^*

Norm conflict resolution algorithm

Given norm system $\mathcal{N} = \{(w_1, \phi_1), \cdots, (w_n, \phi_n)\}$, MDP $\mathcal{M} = \langle S, A, T, R, s_0, \gamma \rangle$:

Before acting ($t = 0$):

- Compute conflict resolution DRAs $\mathcal{D}(\phi_i)$ and product MDP $\mathcal{M}^{\otimes}$
- Graph-theoretic stuff (ensure agent stays in good DRA states)
- Compute $Viol_{\mathcal{N}}$ using value iteration
- Use $Viol_{\mathcal{N}}$ to determine optimal product-space policy π^*

At every time step t, after seeing state s_t:

Norm conflict resolution algorithm

Given norm system $\mathcal{N} = \{(w_1, \phi_1), \cdots, (w_n, \phi_n)\}$, MDP $\mathcal{M} = \langle S, A, T, R, s_0, \gamma \rangle$:

Before acting ($t = 0$):

- Compute conflict resolution DRAs $\mathcal{D}(\phi_i)$ and product MDP $\mathcal{M}^{\otimes}$
- Graph-theoretic stuff (ensure agent stays in good DRA states)
- Compute $Viol_{\mathcal{N}}$ using value iteration
- Use $Viol_{\mathcal{N}}$ to determine optimal product-space policy π^*

At every time step t, after seeing state s_t:

- Use history to figure out current product state $s_t^{\otimes} = (s_t, q_1, \cdots, q_n)$

Norm conflict resolution algorithm

Given norm system $\mathcal{N} = \{(w_1, \phi_1), \cdots, (w_n, \phi_n)\}$, MDP $\mathcal{M} = \langle S, A, T, R, s_0, \gamma \rangle$:

Before acting ($t = 0$):

- Compute conflict resolution DRAs $\mathcal{D}(\phi_i)$ and product MDP $\mathcal{M}^{\otimes}$
- Graph-theoretic stuff (ensure agent stays in good DRA states)
- Compute $Viol_{\mathcal{N}}$ using value iteration
- Use $Viol_{\mathcal{N}}$ to determine optimal product-space policy π^*

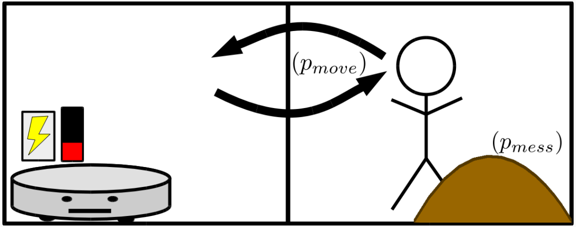
At every time step t, after seeing state s_t:

- Use history to figure out current product state $s_t^{\otimes} = (s_t, q_1, \cdots, q_n)$
- Pick action $a_t = \pi^*(s_t^{\otimes})$

Evaluation: VacuumWorld

- **Scenario 1:** Business as usual
- Norm system:

$$\{(1.0, \mathit{GroomClean})\}$$

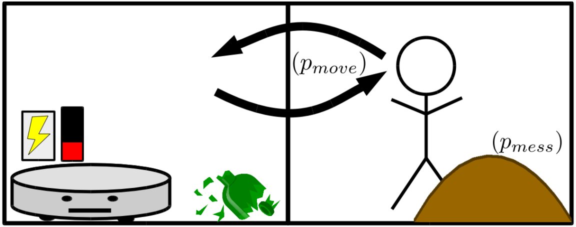


Robot cleaned up messes as quickly as possible (stopping to recharge when necessary)

Evaluation: VacuumWorld

- **Scenario 2:** Broken glass
- Norm system:

$$\{(1.0, \mathbf{G}\,roomClean), (20.0, \mathbf{G}\neg robotDamaged),$$
$$(400.0, \mathbf{G}\neg humanInjured)\}$$

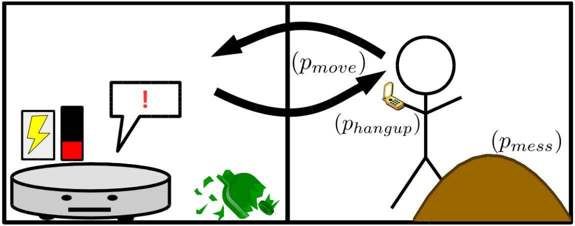


Robot cleaned up glass despite the risk of damage; otherwise as in Scenario 1

Evaluation: VacuumWorld

- **Scenario 3:** Interrupting phone calls
- Norm system:

$$\{(1.0, \mathbf{G}\mathit{roomClean}), (20.0, \mathbf{G}\neg \mathit{robotDamaged}),$$
$$(400.0, \mathbf{G}\neg \mathit{humanInjured}),$$
$$(5.0, \mathbf{G}((\neg \mathbf{X}\mathit{robotSpeak})\ \mathbf{U}\ (\neg \mathit{humanTalking})))\}$$



Robot interrupted phone call (safety > politeness); otherwise as in Scenario 1

Discussion/Future Work

- More sophisticated preference models (e.g., CP-nets)
- Alternatives to discounting violation cost
- More sophisticated logics (e.g., *LDL* - see Brafman, Di Giacomo, and Patrizi 2018; also deontic modality)
- Improving time/space complexity
- Unknown dynamics, POMDPs, multi-agent settings
- Learning norms from natural language (Dzifcak et al. 2009) and from agent behavior (Kasenberg and Scheutz 2017)
    - Inverse norm conflict resolution (Kasenberg and Scheutz 2018)

Acknowledgments

- Funding agencies
  - ONR MURI grant N00014-16-1-2278
  - NSF IIS grant 1723963

References

- Brafman, Ronen, Giuseppe Di Giacomo, and Fabio Patrizi (2018). "LTLf and LDLf Non-Markovian Rewards". In: *Proceddings of the 32nd AAAI Conference on Artificial Intelligence*.
- Ding, Xu Chu et al. (2011). "MDP optimal control under temporal logic constraints". In: *Proceedings of the IEEE Conference on Decision and Control*, pp. 532–538. ISSN: 01912216. DOI: 10.1109/CDC.2011.6161122. arXiv: 1103.4342.
- Dzifcak, Juraj et al. (2009). "What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution". In: *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4163–4168. ISSN: 10504729. DOI: 10.1109/ROBOT.2009.5152776.
- Kasenberg, Daniel and Matthias Scheutz (2017). "Interpretable Apprenticeship Learning with Temporal Logic Specifications". In: *Proceedings of the 56th IEEE Conference on Decision and Control (CDC 2017)*.
- – (2018). "Inverse Norm Conflict Resolution". In: *Proceedings of the 1st AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society*.