Towards Verified Artificial Intelligence

Sanjit A. Seshia
Professor
EECS, UC Berkeley

Joint work with
Dorsa Sadigh, Tommaso Dreossi, Alexander Donze,
Anca Dragan, S. Shankar Sastry

SETTA 2017
October 24, 2017
“Safety critical” systems interacting with humans, often in a complex environment.
Growing Use of Machine Learning/AI in Cyber-Physical Systems

Many Safety-Critical Systems
Artificial Intelligence (AI)

Computational Systems that attempt to mimic aspects of human intelligence, including especially the ability to learn from experience.

How do we ensure that AI-based systems are Dependable?
The **Formal Methods** Lens

- **Formal Methods** ≈ Computational Proof methods
  - Specification/Modeling ≈ Statement of Conjecture/Theorem
  - Verification ≈ Proving/Disproving the Conjecture
  - Synthesis ≈ Generating (parts of) Conjecture/Proof

- Tools/techniques: SAT / SMT solvers, model checkers, theorem provers, simulation-based falsification, ...

**Verification:**

- System S
- Environment E
- Specification ϕ

Does S || E satisfy ϕ?

- YES [+ proof]
- NO [+ counterexample]
Challenges for Verified AI

S. A. Seshia, D. Sadigh, S. S. Sastry.

Does $S \parallel E$ satisfy $\varphi$?

Design Correct-by-Construction instead?

YES [+ proof]

NO [+ counterexample]

Counterexamples, etc. from Rich Signal Spaces?
Talk Outline

• Environment Modeling Challenge
  ➢ Interaction-Aware Control for Human-CPS

• Specification (& Verification) Challenge
  ➢ Verifying Robustness (of Interaction-Aware Controller)
  ➢ Falsification for Deep Learning based CPS

• Conclusions and Future Directions
  – Towards a New Design Methodology for AI-based Systems
Environment Modeling Challenge – Uncertainty and Unknowns

Self-Driving Vehicles: Interact with Humans in Complex Environments; Significant use of machine learning!

Known Unknowns and Unknown Unknowns!!

Cannot represent all possible environment scenarios
Idea 1: **Introspective Environment Modeling**

Impossible to model all possible scenarios

Approach: *Introspect on System to Model the Environment*

Identify: (i) **Interface** between System & Environment, (ii) (Weakest) **Assumptions** needed to Guarantee Safety/Correctness

Algorithmic techniques to *generate weakest interface assumptions* and *monitor them at run-time* for potential violation/mitigation

[Li, Sadigh, Sastry, Seshia; TACAS’14]
Idea 2: **Active Data Gathering and Learning**

*Monitor and Interact with the Environment, Offline and Online, to Model It.*

**Google’s Driverless Cars Run Into Problem: Cars With Drivers**

By MATT RICHTEL and CONOR DOUGHERTY  SEP. 1, 2015

MOUNTAIN VIEW, Calif. —Google, a leader in

“One of the biggest challenges facing automated cars is *blending them into a world in which humans don’t behave by the book.*”
Challenge: Environment (Human) Modeling

Interaction-Aware Control

Lane Change on a Highway
Interaction as a Dynamical System

\[ x^{t+1} = f_H(f_R(x^t, u^t_R), u^t_H) \]

Robot actions \( u_R \)

Human actions \( u_H \)

Model the problem as a *Stackelberg (turn-based) Game*. Robot moves first.
Assumptions/Simplifications

Model Predictive (Receding Horizon) Control:

Optimize over short time horizon $N$, replan at every step $t$.

$$R_R(x, u_R, u_H) = \sum_{t=1}^{N} r_R(x^t, u_R^t, u_H^t)$$

$$R_H(x, u_R, u_H) = \sum_{t=1}^{N} r_H(x^t, u_R^t, u_H^t)$$

Assume deterministic “rational” human model, human optimizes reward function which is a linear combination of “features”.

Human has full access to $u_R$ for the short time horizon.

$$u^*_H(x_0, u_R) = \arg\max_{u_H} R_H(x_0, u_R, u_H)$$
Learning (Human) Driver Models

Learn Human’s reward function based on Inverse Reinforcement Learning [Ziebart et al, AAAI’08; Levine & Koltun, 2012].

Assume structure of human reward function:

\[ r_H(x^t, u_R^t, u_H^t) = w^\top \phi(x^t, u_R^t, u_H^t) \]

(a) Features for the boundaries of the road  
(b) Feature for staying inside the lanes.  
(c) Features for avoiding other vehicles.

Interaction as a Dynamical System

Find optimal actions for the autonomous vehicle while accounting for the human response $u_H^*$. 

\[ u_R^* = \underset{u_R}{\text{argmax}} \ R_R(x_0, u_R, u_H^*(x_0, u_R)) \]

Model $u_H^*$ as optimizing the human reward function $R_H$.

\[ u_H^*(x_0, u_R) = \underset{u_H}{\text{argmax}} \ R_H(x_0, u_R, u_H) \]
Solution of Nested Optimization

\[ u_R^* = \arg\max_{u_R} R_R(x, u_R, u_H^*(x, u_R)) \]

\[ R_R(x, u_R, u_H) = \sum_{t=1}^{N} r_R(x^t, u_R^t, u_H^t) \]

**Gradient-Based Method (Quasi-Newton):** (solve using L-BFGS technique)

\[ \left\{ \begin{array}{l}
R_R(x, u_R, u_H^*) \\
\frac{\partial R_R}{\partial u_R} = \frac{\partial R_R}{\partial u_H} \frac{\partial u_H^*}{\partial u_R} + \frac{\partial R_R}{\partial u_R}
\end{array} \right. \]

\[ u_H^*(x, u_R) \approx \arg\max_{u_H} R_H(x, u_R, u_H) \]

\[ R_H(x, u_R, u_H) = \sum_{t=1}^{N} r_H(x^t, u_R^t, u_H^t) \]
Cautious Lane Change
Interaction-Aware Lane Change
We can’t rely on a single driver model.

We need to differentiate between different drivers.
\[ p(u_H | x) \propto \exp(R_H(x, u_H)) \]
\[ p(u_H \mid x, \theta) \propto \exp(R_H(x, u_H, \theta)) \]

\[ b_{t+1}(\theta) \propto b_t(\theta) \cdot p(u_H \mid x_t, \theta) \]
\[ p(u_H | x, \theta) \propto \exp(\mathcal{R}_H (x, u_H, \theta)) \]

\[ b_{t+1}(\theta) \propto b_t(\theta) \cdot p(u_H | x_t, \theta) \]
\[ p(u_H | x, \theta, u_R) \propto \exp(R_H(x, u_H, \theta, u_R)) \]

\[ b_{t+1}(\theta) \propto b_t(\theta) \cdot p(u_H | x_t, \theta, u_R) \]

\[ u_R = \arg\max_{u_R} R_R \]
\[ p(u_H | x, \theta, u_R) \propto \exp(R_H(x, u_H, \theta, u_R)) \]

\[ b_{t+1}(\theta) \propto b_t(\theta) \cdot p(u_H | x_t, \theta, u_R) \]

**Info Gathering**

\[ R_R(x, u_H, \theta, u_R) = \mathbb{H}(b_t) - \mathbb{H}(b_{t+1}) + \lambda \cdot R_{\text{goal}}(x, u_H, \theta, u_R) \]

**Goal**

\[ u_R = \arg \max_{u_R} \mathbb{E}_\theta [R_R] \]
Nudging in for Active Info Gathering
Nudging in for Active Info Gathering
Distracted Human Driver
Key Ideas:
Actively gather data about the environment (human) by affecting the environment’s behavior

Learn environment (human) model from data, update online

Questions:
• How to verify such human-robot systems?
• What are more realistic human models? (e.g. “bounded rationality”)
Talk Outline

• Environment Modeling Challenge
  ➢ Interaction-Aware Control for Human-CPS

• Specification (& Verification) Challenge
  ➢ Verifying Robustness (of Interaction-Aware Controller)
  ➢ Falsification for Deep Learning based CPS

• Conclusions and Future Directions
  – Towards a New Design Methodology for AI-based Systems
More Efficient, but is it Safe?
Verifying Temporal Logic Requirements

**Signal Temporal Logic (STL)** [Maler & Nickovic, ‘04]

Predicates over continuous signals, Propositional Formulas $\varphi$ ($\land, \lor, \neg$ of the predicates), Temporal Operators ($G, F, X, U$), real-time interval $\tau$.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_\tau \varphi$</td>
<td>$\varphi$ is true at all future moments in $\tau$.</td>
</tr>
<tr>
<td>$F_\tau \varphi$</td>
<td>$\varphi$ is true in some future moment in $\tau$.</td>
</tr>
<tr>
<td>$\varphi_1 U_\tau \varphi_2$</td>
<td>$\varphi_1$ is true until $\varphi_2$ becomes true in $\tau$.</td>
</tr>
</tbody>
</table>

**Safety** (invariance): Vehicle maintains specified distance from obstacles.

$$G_{[0,\tau]} \left[ \text{dist(vehicle, obstacle) > } \Delta \right]$$
From Logical Formulas to Objective Functions

• STL formula has both
  – Boolean semantics: true/false
  – Quantitative semantics: value in $\mathbb{R}$

• Example:

$$G_{[0,\tau]}(\text{dist(vehicle, obstacle)} > \Delta)$$

$$\inf_{[0,\tau]} [\text{dist(vehicle, obstacle)} - \Delta]$$
\[ u_H^* = \arg\max_{u_H} R_H(x, u_R, u_H^*) \]

\[ u_R^* = \arg\max_{u_R} R_R(x, u_R, u_H^*(x, u_R)) \]
$R_H(x, u_H, u_R)$
$R^\dagger_{\mathcal{H}}(x, u_{\mathcal{H}}, u_R)$

$|R^\dagger_{\mathcal{H}} - R_{\mathcal{H}}| < \delta$
How robust is the learning-based controller?

How to algorithmically find falsifying actions by the human?
\[ \overline{u_H} = \arg \min_{u_H} R_R(x, u^*_R, u_H) \]

\[ \text{s.t. } \exists R^\dagger_H : u_H = \arg \max_{u_H} R^\dagger_H(x, u^*_R, \overline{u_H}) \]

\[ |R^\dagger_H - R_H| < \delta \]

Falsifying actions

Optimizing a perturbed version of the learned reward function.
Theorem:

\[ \overline{u_H} = \arg \min_{u_H} R_R(x, u_R^*, u_H) \]
\[ |R_H^\dagger - R_H| < \delta \]
\[ \text{s.t. } \exists R_H^\dagger : u_H = \arg \max_{\overline{u_H}} R_H^\dagger (x, u_R^*, \overline{u_H}) \]

\[ \overline{u_H} = \arg \min_{u_H} R_R(x, u_R^*, u_H) \]
\[ \text{s.t. } R_H(x, u_R^*, u_H) > R_H(x, u_R^*, u_H^*) - 2\delta \]

\[ \overline{u_H} = \arg \min_{u_H} R_R(x, u^*_R, u^*_H) \]

\[ \text{s.t. } R_H(x, u^*_R, u^*_H) > R_H(x, u^*_R, u^*_H) - 2\delta \]
\[ |R_{\mathcal{H}}^+ - R_{\mathcal{H}}| \leq \delta \]

\[ \delta = 0 \]
\[ |R_H^\dagger - R_H| < \delta \]
\[ \delta = 0.025 \]
\[ |R_{\mathcal{H}}^+ - R_{\mathcal{H}}| < \delta \]

\[ \delta = 0.15 \]
\[ \Pr(|R^+_H - R_H| < \delta) > 0.9 \]
\[ \delta = 0.15 \]
Key Ideas:

Turn Verification (falsification) into Optimization

Important Property: Robustness of AI/Learning-based system to small perturbations in data/learned function
Falsification of Cyber-Physical Systems with Machine Learning Components

Problem: Verify Automotive System (CPS) that uses ML-based Perception

- **Focus:**
  - **Falsification:** finding scenarios that violate safety properties
  - **Test (Data) Generation:** generate “interesting” data for training / testing → improve accuracy
  - **Deep Neural Networks**, given the increasing interest and use in the automotive context.
Automatic Emergency Braking System (AEBS)

Deep Learning-Based Object Detection

• Goal: Brake when an obstacle is near, to maintain a minimum safety distance
  • Controller, Plant, Env models in Matlab/Simulink
  • Object detection/classification system based on deep neural networks
    • Inception-v3, AlexNet, ... trained on ImageNet
What’s the Specification for Perception Tasks?

Convolutional Neural Network trained to recognize cars

How do you formally specify “a car”?
Idea: Use a **System-Level Specification**

- **✗** “Verify the Deep Neural Network”
- **✓** “Verify the System containing the Deep Neural Network”

Formally Specify the *End-to-End Behavior* of the System

STL Formula: \( G \left( dist(ego \text{ vehicle}, \text{env object}) > \Delta \right) \)
Tool: Simulation-Based Falsification of Signal Temporal Logic for CPS

• STL has quantitative semantics
  – Logical formula $\rightarrow$ Cost Function $\rho$
  – Quantifies “how much” a trace satisfies a property

• Advantage: Finding a bug (property violation) corresponds to minimizing the function $\rho$ and checking if the value falls below 0.
  – This view of “verification as optimization” underlies the Breach toolkit and similar tools
Our Approach: Combine Temporal Logic CPS Falsifier with ML Analyzer

- CPS Falsifier uses abstraction of ML component
  - Optimistic analysis: assume ML classifier is always correct
  - Pessimistic analysis: assume classifier is always wrong
- Difference is the region of interest where output of the ML component “matters”

Compositional:
CPS Falsifier and ML Analyzer can be designed and run independently (& communicate)!
Identifying Region of Interest for Automatic Emergency Braking System

ML always correct  ML always wrong  Potentially unsafe region depending on ML component (yellow)

Perform Optimistic and Pessimistic Analyses on the Deep Neural Network
Machine Learning Analyzer

Systematically Explore Region of Interest in the Image (Sensor) Space

Feature space $\tilde{X}$

Systematic Sampling (low-discrepancy sampling)

Abstract space $A$

Neural network $y \in \{\text{car}, \neg \text{car}\}$
Sample Result

Inception-v3 Neural Network
(pre-trained on ImageNet using TensorFlow)

This misclassification may not be of concern
Sample Result

*Inception-v3*

*Neural Network*

*(pre-trained on ImageNet using TensorFlow)*

Corner case Image

Misclassifications

But this one is a real hazard!
Newer Results

[Dreossi, Ghosh, et al., ICML 2017 workshop]

Results on squeezeDet NN and KITTI dataset for autonomous driving
Summary of Key ideas

• Generate adversarial examples that violate system-level specification

• Compositional Approach blends the strengths of the CPS Falsifier with a Machine Learning Analyzer

• Counterexample images can be added to the training set to improve ML accuracy (“right” data vs. “big” data)

• Ongoing/Future Work:
  – Improving ML analyzer
  – New benchmarks (datasets and networks)
  – Evaluating training/test accuracy improvements
Concluding Thoughts
# Towards Verified Artificial Intelligence

<table>
<thead>
<tr>
<th>Challenges</th>
<th>Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Environment (incl. Human) Modeling</td>
<td>Data-Driven, Introspective Environment Modeling</td>
</tr>
<tr>
<td>3. Learning Systems</td>
<td>Abstract &amp; Explain</td>
</tr>
<tr>
<td>Complexity</td>
<td></td>
</tr>
<tr>
<td>4. Efficient Training, Testing, Verification</td>
<td>Verification-Guided, Adversarial Analysis and Improvisation</td>
</tr>
<tr>
<td>5. Design for Correctness</td>
<td>Formal Inductive Synthesis</td>
</tr>
</tbody>
</table>

Correct-by-Construction Design with Formal Inductive Synthesis

Inductive Synthesis: Learning from Examples (ML)

Formal Inductive Synthesis: Learn from Examples while satisfying a Formal Specification

Key Idea: **Oracle-Guided Learning**
Combine Learner with Oracle (e.g., Verifier) that answers Learner’s Queries

Verifier-Guided Training of Deep Neural Networks

• Instance of Oracle-Guided Inductive Synthesis
• Oracle is Verifier (CPSML Falsifier) used to perform counterexample-guided training of DNNs
• Substantially increase accuracy with only few additional examples
# Towards **Verified Artificial Intelligence**

## Challenges

1. Environment (incl. Human) Modeling
2. Specification
3. Learning Systems Complexity
4. Efficient Training, Testing, Verification
5. Design for Correctness

## Principles

- Data-Driven, Introspective Environment Modeling
- System-Level Specification; Robustness/Quantitative Spec.
- Abstract & Explain
- Verification-Guided, Adversarial Analysis and Improvisation
- Formal Inductive Synthesis

---

**Exciting Times Ahead!!! Thank you!**

---