Explaining Errors in Complex Systems

A Diagnosis Tool and Testing Framework for Robust Decision Making

Leilani H. Gilpin, Assistant Professor
Dept. of CSE, UC Santa Cruz
CSE 200
November 18, 2021
Agenda

Motivate problem: Autonomous Vehicles are Prone to Failure

Anomaly Detection through Explanations (ADE): a Diagnosis Tool for AVs.

Future work: Explainable Tasks for Robust and Secure Hybrid Systems.

Question: How to develop self-explaining architectures that can help anticipate failures instead of after-the-fact?
Complex Systems Fail in Complex Ways

Autonomous Vehicle Solutions are at Two Extremes

Serious safety lapses led to Uber’s fatal self-driving crash, new documents suggest

Problem: Need better sanity checks and communication

My Herky-Jerky Ride in General Motors' Ultra-Cautious Self Driving Car

GM and Cruise are testing vehicles in a chaotic city, and the tech still has a ways to go.
Architecture Inspired by Human Organizations

Communication and Sanity Checks

1. Hierarchy of overlapping committees.
2. Continuous interaction and communication.
3. When failure occurs, a story can be made, combining the members’ observations.
An Architecture to Mitigate Common Problems

Synthesizer to reconcile inconsistencies between parts.

Local Sanity Checks

Reconcile conflicting reasons. Justify new examples.
An Existing Problem
The Uber Accident
The best option is to veer and slow down. The vehicle is traveling too fast to suddenly stop. The vision system is inconsistent, but the lidar system has provided a reasonable and strong claim to avoid the object moving across the street.
Motivate problem: Autonomous Vehicles are Prone to Failure

Anomaly Detection through Explanations (ADE): a Diagnosis Tool for AVs.

Future work: Explainable Tasks for Robust and Secure Hybrid Systems.
Limited Internal Reasoning

A Google self-driving car caused a crash for the first time

A bad assumption led to a minor fender-bender

Serious safety lapses led to Uber’s fatal self-driving crash, new documents suggest

My Herky-Jerky Ride in General Motors' Ultra-Cautious Self Driving Car

GM and Cruise are testing vehicles in a chaotic city, and the tech still has a ways to go.
Reconciling Internal Disagreements
With an Organizational Architecture

- Monitored subsystems combine into a system architecture.
- Explanation synthesizer to deal with *inconsistencies*.
  - Argument tree.
  - Queried for support or counterfactuals.

Anomaly Detection Through Explanations
Anomaly Detection through Explanations
Reasoning in Three Steps

1. Generate Symbolic Qualitative Descriptions for each committee.
2. Input qualitative descriptions into local “reasonableness” monitors.
3. Use a synthesizer to reconcile inconsistencies between monitors.
3. Use a synthesizer to reconcile inconsistencies between monitors.

- Explanation synthesizer to deal with inconsistencies.
- Argument tree.
- Queried for support or counterfactuals.

Synthesizer + Priority Hierarchy → Abstract Goals

1. **Passenger Safety**
2. Passenger Perceived Safety
3. Passenger Comfort
4. Efficiency (e.g. Route efficiency)

A passenger is safe if:

- The vehicle proceeds at the same speed and direction.
- The vehicle avoids threatening objects.
Use a synthesizer to reconcile inconsistencies between monitors.

\[(\forall s, t \in \text{STATE}, v \in \text{VELOCITY}) \]
\[((\text{self, moving}, v), \text{state}, s) \land ((t, \text{isSuccessorState}, s)) \land ((\text{self, moving}, v), \text{state}, t) \land (
\forall x \in \text{OBJECTS} \text{ s.t. } ((x, \text{isA}, \text{threat}), \text{state}, s) \lor ((x, \text{isA}, \text{threat}), \text{state}, t)))\]

\[\Rightarrow (\text{passenger, hasProperty, safe})\]

\[(\forall s \in \text{STATE}, x \in \text{OBJECT}, v \in \text{VELOCITY}) \]
\[((x, \text{moving}, v), \text{state}, s) \land ((x, \text{locatedNear}, \text{self}), \text{state}, s) \land ((x, \text{isA}, \text{large_object}), \text{state}, s) \]
\[\Leftrightarrow ((x, \text{isA}, \text{threat}), \text{state}, s))\]

A passenger is safe if:

- The vehicle proceeds at the same speed and direction.
- The vehicle avoids threatening objects.
3. Use a synthesizer to reconcile inconsistencies between monitors.

\[ (\forall s, t \in \text{STATE}, v \in \text{VELOCITY} \]
\[ \left( (\text{self}, \text{moving}, v), \text{state}, s \right) \land \]
\[ (t, \text{isSuccessorState}, s) \land \]
\[ \left( (\text{self}, \text{moving}, v), \text{state}, t \right) \land \]

\[ (\exists x \in \text{OBJECTS} \ s.t. \]
\[ \left( (x, \text{isA}, \text{threat}), \text{state}, s \right) \land \]
\[ \left( (x, \text{isA}, \text{threat}), \text{state}, t \right) \]

\[ \Rightarrow (\text{passenger}, \text{hasProperty}, \text{safe}) \]
3. Use a synthesizer to reconcile inconsistencies between monitors.

**Abstract Goal Tree**

'passenger is safe',
AND(
  'safe transitions',
  NOT('threatening objects')
)

**List of Rules**

IF ( AND('moving (?v) at state (?y)',
  '(?z) succeeds (?y)',
  'moving (?v) at state (?z)'),
  THEN('safe driving at (?v) during (?y) and (?z)'))

IF (OR('obj is not moving',
  'obj is not located near',
  'obj is not a large object'),
  THEN('obj not a threat at (?x)'))

IF (AND('obj not a threat at (?y)',
  'obj not a threat at (?z)',
  '(?z) succeeds (?z)'),
  THEN('obj is not a threat between (?y) and (?z)'))

**Backwards Chain**

passenger is safe at V between s and t
AND (AND (moving V at state s
  t succeeds s
  moving V at state t )
  AND (OR ( obj is not moving at t
    obj is not locatedNear at t
    obj is not a large object at t ))
  OR ( obj is not moving at t
    obj is not locatedNear at t
    obj is not a large object at t )))
The best option is to veer and slow down. The vehicle is traveling too fast to suddenly stop. The vision system is inconsistent, but the lidar system has provided a reasonable and strong claim to avoid the object moving across the street.
Uber Example in Simulation

Evaluation of Error Detection is Difficult

Real-world Inspired Scenarios

- **Detection**: Generate logs from scenarios to detect failures.
- **Insert errors**: Scrambling *multiple* labels on existing datasets.
- **Real errors**: Examining errors on the validation dataset of NuScenes leaderboard.

Reconcile Inconsistencies
Approach: Content Generation

Anticipatory Thinking Layer for Error Detection

DALL-E Generates “A chair in the shape of an avocado”
Approach: Content Generation
Anticipatory Thinking Layer for Error Detection

Generate images with shadows before tunnels.

Generate images with fallen signs.

Generate images with trucks carrying traffic lights.
Motivate problem: Autonomous Vehicles are Prone to Failure

Anomaly Detection through Explanations (ADE): a Diagnosis Tool for AVs.

Future work: Explainable Tasks for Robust and Secure Hybrid Systems.
Hybrid Systems with Humans and Machines
Working Together on Shared Tasks

Explanations are a debugging language.

- Debugging: humans can improve complex systems.
- Education: complex systems can “improve” or teach humans.
Ex post facto explanations

- Log data
- Input
- Sensor data

Learning system → “Explanation” → Debugging
The object to the left is 5 ft tall, moving towards the right. It’s the salient feature of neurons. Contextual justification: “This is a person because they have the right shape and movement.”

Humans

Feedback
Validation
Help on tasks

Security

Game adversaries

Dev testing

Learning system

Saliency map

Symbolic system

Input

Log data

Sensor data

Learning system

Feedback
Validation
Help on tasks

Humans

Contextual justification: “This is a person because they have the right shape and movement.”
Explaining Errors in Complex Systems

Explanations and Reasons that Society can **Trust**
- Systems that can **testify**, answer questions, and **provide insights**.
- Systems that use **commonsense**, similar to the ways that humans do.

A Common Language for Debugging and **Diagnosis**
- Interactive tools using explanations as a common **debugging language**.
- Systems that **articulately communicate with humans** on shared tasks.

Articulate Mechanisms that are **Robust**
- Hybrid, symbolic, learning systems that solve problems in **multiple ways**.
- **Dynamic explanations**, under uncertainty for safety or mission-critical tasks.

---

lgilpin@ucsc.edu
Leilani H. Gilpin

http://lgilpin.com
UC Santa Cruz