Explaining Possible Futures for Robust Autonomous Decision-Making

Leilani H. Gilpin
MIT CSAIL
lgilpin@mit.edu

Abstract

As humans, we consider multiple alternatives when making decisions or choices. The way we decide between these choices is to create stories and explain (to ourselves) why they are reasonable or not. But when machines make decisions, their processes are not interpretable (understandable by humans) nor explainable (able to recount the reasons and dependencies leading up to a decision). In this paper, I present a methodology to *explain* possible futures and utilize these explanations to make more robust and reasonable decisions moving forward. Internal explanations will be used dynamically by the parts of a complex machine to detect failure and intrusion. System-level explanations will provide a coherent and convincing story to humans for engineering, legal reasoning, and forensics.

Introduction

Making decisions is a difficult human task. Often times, we are left wondering if we made the *correct* decision (usually, after the fact). These decisions may be due to overestimating, underestimating or miscalculating the impact; we may not have *explained* the alternatives accurately. Other times, we may have had a time constraint, leading to an impulsive decision without properly processing the outcomes and alternatives.

Explaining possible futures is important as autonomous agents are increasingly deployed in real world-settings (e.g. driving), where there has been an increase in malfunctions and errors leading to injuries[1] and even deaths[2]. Such level of increased harm on human lives is undesirable and completely untenable. My research addresses the uncertain, unstable, and error-prone decision-making of complex machine by imposing *explanations*: the symbolic reasons, premises, and support leading up to an intended decision.

[1]Mall robot injures a toddler: https://qz.com/730086/a-robot-mall-cop-did-more-harm-than-good/

[2]Uber self-driving car pedestrian fatality: https://www.nytimes.com/interactive/2018/03/20/us/self-driving-uber-pedestrian-killed.html

The long-term goal for autonomous vehicles is to minimize the numbers of false positive and false negative detection so that human fatalities become rare. In the meantime, explanations can be used to make the subsystems accountable and learn from their mistakes. For example, there are two ways in which a previous working system can exhibit anomalous behavior. A local error is confined to a particular subsystem. An example of which is a subsystem that is calculating square roots, but the output squared is not "reasonably close" to the input. The second type of inconsistency is observed as a failed cooperation between subsystems; each subsystem is able to defend its observed behavior, but the larger neighborhood of subsystems is not executing its shared task as intended. For example, take a mechanism that is solving an optimization problem. Within this mechanism, there is a neighborhood of subsystems with the common goal of calculating the next step gradient for each successive iteration. Within this neighborhood, there is a subsystem whose job is to calculate square roots. The square root subsystem is only returning the positive square root. This output is not inconsistent local to the subsystem: it is able to explain that its behavior is reasonable since the output squared is close to input. However, when cooperating with other subsystems in its community, one or more subsystems is expecting the conjugate root, and the community of cooperating subsystems exhibits unexpected behavior. Although each individual subsystem can explain that it is behaving reasonably, this community of subsystems needs additional help to ameliorate the inconsistencies in neighborhoods of interconnected subsystems and develop an explanation.

I have developed a proof-of-concept that can make these kinds of deductions using explanations. This methodology is focused on identifying, detecting, and explaining anticipatory subsystem decisions. In future work, the explanations will be processed *automatically* to determine the "best" next step. In this paper, I define the problem space, present a proof-of-concept with initial results, and propose a methodology for anomaly detection and monitoring with explanations.

Not all successful systems will make decisions this way. But, this methodology will serve as an inspiration for the development of machines that have to make safety-critical or mission-critical decisions.

Problem Statement

Complex systems have become a staple of daily life, providing everything from smart thermostats to online banking to collision-avoidance systems. In a perfect world, these systems would have no errors, and false positives would be nearly impossible. Although these systems can be tested in simulation, simulation environments cannot contain all the factors in the real world that can invoke errors and anomalies. At the same time, test suites of prone error modes cannot possibly represent all possible error cases. Although simulation and test cases can represent and catch a multitude of error conditions, we need better error detection and explanation protocols in practice.

Tools and frameworks exist that permit the design and creation of some systems that are provably correct by construction. Unit testing is a well-accepted practice for verifying the behavior of a system once it is realized. However, these approaches are ultimately inadequate, as the specifications to which systems are constructed are complex, subject to error, and constantly evolving in response to shifting requirements. As these systems become larger, containing more interacting subsystems handling a wider range of tasks, the number of possible failure modes increases, so we can expect this already challenging problem to grow worse over time.

Instead of striving to produce a perfect design that never fails I propose to build in mechanisms that robustly detect failures and attempt to ameliorate their own anomalies through explanation. If part of a system produces undesirable behavior, either as an intrinsic error or as a result of external interference, the rest of the system should be able to dynamically limit the extent of the possible damage.

The goal of this methodology is three-fold: to correctly identify (minimize false positives), detect (using constant introspection and monitoring) and explain subsystem failures and alternatives.

Imagining the Future with Explanations

Sound decisions are not made based on some single instant in time; rather, they are made with careful consideration of their consequences. Sometimes these consequences may not be known beforehand. This was stated nicely by Donald Rumsfeld:

> There are known knowns. These are things we know that we know. There are known unknowns. That is to say, there are things that we know we don't know. But there are also unknown unknowns. There are things we don't know we don't know.

Thus, a system must be able to imagine each possible future that may result from its choices, and evaluate whether that future might be reasonable. To do this well, the system must be able to *simulate* the behavioral and physical consequences of acting on any set of premises that may be chosen by committee arbitration, particularly in the case where it will have accepted premises that in fact represent the wrong situation.

Given a set of premises whose relative validity may not be obvious, it can be difficult to decide which premises to accept. There are of course situations in which premises can be identified as faulty – for example, if they lead to violations of reasonable constraints or contradictions of systematic axioms. Often, though, we cannot reject any of the available premises outright. In such cases we need a way to decide which system of premises, out of equally plausible-looking possibilities, is to be accepted in practice – and, as a corollary, what future our system pursues.

One can try to treat this problem as one of simulation-based *search*, with a possible objective being to find the choice of premises resulting in the "least bad" set of consequences given the set of possible realities under consideration. It is of course impossible to consider every potentially available reality; for instance, one cannot, in this process, reasonably account for events like a meteor strike during the interval under consideration, where no evidence exists to support the notion of incoming meteors. However, information reported by onboard perception supporting the existence of an "unusual" object, like a lawnmower, provides at least two possibilities – the existence of a lawnmover, or a defect in the onboard perception – whose consequences must be examined.

Commonsense Reasoning

Humans are opaque systems. When something goes wrong, we cannot always say why. For example, when we are ill or malfunctioning, we cannot always point to the exact subsystem causing the error. But we can form a coherent explanation of what we *believe* we are suffering from, by querying previous data and commonsense. If we have a fever, we can usually come up with a reason: we feel hot, then cold, and that is *similar to* a previous time when we had a fever. We can also create explanations with commonsense. If we have a stomach ache, perhaps it was the spicy food that caused the pain. Or, it was the fact that we ate a heavy meal on a previously [starved] stomach.

One way to mitigate perception errors is to supplement decisions with commonsense. One way to do this is to impose a *reasonableness monitor* (Gilpin 2018). For example, in the stomach ache example, we can use commonsense to come up with multiple explanations. This requires the availability of a commonsense knowledge base. We can formulate explanations using "nearby" information: the stomach is close to the appendix, which may be ruptured. Or we can create other causal explanations: stomach aches can be caused by spicy food, or stomach aches can be caused by eating too much on an empty stomach. It is difficult to determine which one of these explanations is "most" correct or plausible, which is left to future work. But, the ability for intelligent machines to use commonsense to formulate these explanations themselves is a promising area of research.

A Preliminary Demonstration

To demonstrate how explanations could be used to imagine possible futures, I constructed a small proof-of-concept demonstration. Consider a toy model of a car, consisting of low-level actuation components, like the braking, steering, and power control systems, as well as driving tactics, LiDAR and the vision components, as seen in Figure 4

There are monitors around each component, and a high-level reasoner to reconcile component explanations for higher-level decision making. The high-level reasoner takes in the input from the three underlying components, and proposes a few candidate high-level decisions. This high-level reasoner examines these proposed plans along with the explanations from the underlying parts to make a more informed, explainable, and robust plan. The system also includes a priority hierarchy to enforce individual needs when there are conflicts. For example, the vehicle's inhabitant(s) are prioritized, then other drivers and pedestrians, etc.

Scenario Information

The example for the proof-of-concept is the Uber self-driving vehicle accident. On March 18, 2018 at approximately 10pm, an Uber Advanced Technologies Group (ATG) self-driving test vehicle (a modified 2017 Volvo XC90) struck and killed a pedestrian in Tempe, Arizona. In the investigation findings: "The Uber ATG automated driving system detected the pedestrian 5.6 seconds before impact. Although the system continued to track the pedestrian until the crash, it never accurately identified the object crossing the road as a pedestrian – or predicted its path[3]."

Although the LiDAR system had correctly detected the pedestrian, since the vision system was unreliable, the planning system was instructed to ignore[4] the detected object as a false positive, and continued forward at a high speed. This error is due to an *inconsistency between parts* and an inability to anticipate consequences. I can reconcile this inconsistency using internal, subsystem explanations and a set of future plans. These are the facts from the initial Uber report[5].

1. Radar and LiDAR detected the pedestrian about 6 seconds before impact (vehicle speed was 43 mph).
2. The vision system classified the pedestrian as an unknown object, as a vehicle, and then as a bicycle with varying expectations of future travel path.
3. 1.3 seconds before impact, the self-driving system engaged an emergency braking maneuver, to mitigate a collision.

Proof of Concept

Consider the Uber scenario approximately 6 seconds before impact. The high-level reasoner (or a monitor around the will generate 3 plans with some certainty. The high-level reasoner will explain these high-level plans as follows:

[3]NTSB Accident Report Press Release-https://ntsb.gov/news/press-releases/Pages/NR20191119c.aspx

[4]Uber data inconsistency: https://www.theinformation.com/articles/uber-finds-deadly-accident-likely-caused-by-software-set-to-ignore-objects-on-road

[5]NTSB Preliminary report-https://www.ntsb.gov/investigations/AccidentReports/Reports/HWY18MH010-prelim.pdf

```
This vision perception is unreasonable.
There is no commonsense data supporting
the similarity between a bike, vehicle
and unknown object except that they can
be located at the same location.
This component should be ignored.
```

Figure 1: The output of a local reasonableness monitor on the input from the Uber self-driving car scenario, in which the vision system was oscillating between 3 labels: a bike, a vehicle, and an unknown object. The perception is classified as unreasonable.

```
This LiDAR perception is reasonable.
An object moving of this size is a large
moving object that should be avoided.
```

Figure 2: The output of a local reasonableness monitor on the LiDAR input from the Uber self-driving car scenario. Since a large object is detected, the monitor recommends it to be avoided.

1. Continue straight.
2. Slow down to a stop.
3. Veer to the side of the road.

Note that these intended decisions are not necessarily output in this human-understandable way. But using edge-detection and interval analysis with explanations which was explored in previous work (Gilpin and Yuan), I can directly generate these kinds of text explanations from symbolic descriptions.

Now, the high-level reasoner also requires explanations from its underlying parts. In this scenario, the high-level reasoner receives input from the computer vision (perception) system, the LiDAR/radar system, and the driving tactics (consisting of the brakes, steering, gas, etc.) The system-wide monitoring diagram for this example is shown in Figure 4. The vision system output is a set of segmentations and their corresponding labels (e.g. person, tree, road, etc.) For this Uber example case, I focus on the segmentation in the upper left (from the point of view of the car). In the seconds before impact, the output of the reasonableness monitor for the vision processing component is shown in Figure 1.

But there is more sensory information: the LiDAR sensor data log. The LiDAR reasonableness monitor first *interprets* the sensor log. Using edge detection and interval analysis, the raw sensor data is abstracted into a list of symbolic descriptions that can be passed into the reasonableness monitor. The symbolic list produced for the LiDAR data in this scenario is `('object, 'moving, '5-ft-tall, 'top-left-quadrant, ...)`. In the seconds before impact, the output of the reasonableness monitor for the vision processing component is shown in Figure 2.

Finally, the tactics system is similarly interpreted into a symbolic, qualitative description: `('moving-quickly, 'straight, 'continued-straight, ...)` signifying that the vehicle has been proceeding straight, quickly for the last 5-10 second horizon. The reasonableness

```
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.
```

Figure 3: The high-level reasoner output for the Uber self-driving vehicle example.

monitor for the tactics system deduces that that system state is reasonable: `The system state is reasonable given that the vehicle has been moving quickly and proceeding straight for the last 10 second history.`

With these three subsystem explanations, the high-level reasoner processes the explanations (which are also stored as a list of symbolic triples). The reasoner examines and assesses at the strengths of each explanation, and compares it to a hierarchy of needs to see which intended decision does not violate the the needs hierarchy. Several iterations of this process may be necessary for more complex decisions (or a more complicated needs hierarchy). For this proof-of-concept, the high-level reasoner explains each of the intended plans against the component explanations and hierarchy of needs:

1. Continue forward (straight): this would result in injuring the object detected by the LiDAR system. The vision system cannot confirm this detection and is deemed unreliable due to misjudgements. Therefore, the vehicle should not continue forward.
2. Stop: It is unclear if stopping would guarantee limited harm to the object detected by the LiDAR system. A sudden stop at the speed of the vehicle may injure its occupants. Therefore, the vehicle should not stop. (Although, this intended decision will remain a possible choice since it does not produce as much damage as the first option).
3. Veer and slow down: this would result in avoiding the object detected by the LiDAR system. The vision system cannot confirm this detection and is deemed unreliable due to misjudgements. This is consistent to safely avoid the object. Veering and slowing down causes less damage to the vehicle occupants.

And the final explanation produced by the high-level reasoner is shown in Figure 3.

Previous Work

One goal of this work is to decrease the number of false positives in anomaly detection by using explanations. Anomaly detection is a well-studied area in data science and machine learning (Chandola, Banerjee, and Kumar 2009), even as a tactic to combat intrusion detection in networks (Garcia-Teodoro et al. 2009). In developing anomaly detection for autonomous systems, it is also necessary to develop real-time anomaly detection algorithms. Real-time anomaly detection needs a novel scoring algorithm designed for streaming data, including a series of benchmarks (Lavin and Ahmad 2015). However, decreasing the number of false-positives and false-negatives in anomaly detection is a difficult problem. Some tactics include smoothing the output (Grill, Pevnỳ, and Rehak 2017), or piece-wise approximations (Vallis, Hochenbaum, and Kejariwal 2014).

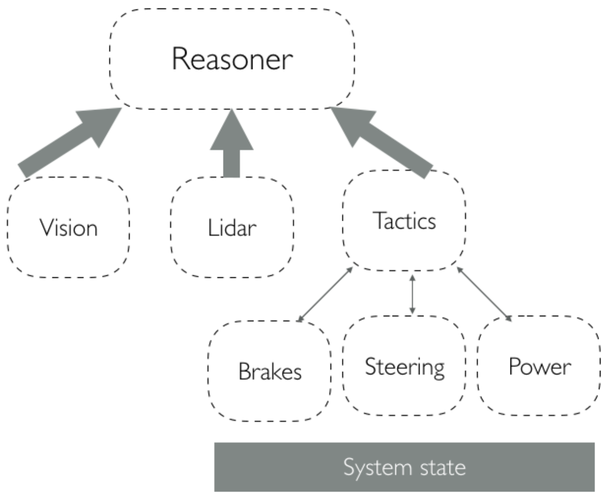

Figure 4: System diagram for the explanatory architecture for a simplified self-driving car. Tactics provide communication to and from the brakes, gas and power subsystems. The tactics system reports its reasons and explanations to the reasoner, as well as the LiDAR and vision subsystems.

Another goal is to make autonomous systems naturally resistant to intrusion through monitoring and continuous introspection. Intrusion detection research is a lively field with numerous proposed approaches in the literature. Some approaches rely on a combination of topological vulnerability analysis and system alert data to detect attacks (Albanese et al. 2011). Other approaches are specifically for collections of autonomous flying vehicles, directly examine deviations from expected control algorithm behavior to detect faulty agents and route communication around them(Negash, Kim, and Choi). The primary deficiency of these approaches is that, while they provide fault detection, they do not attempt any explanation of how the faults might have arisen, a deficiency we hope to address through our work.

The final goal of this work is to use provide interpretable explanations. Within the context of self-driving car, previous work has used reasoning systems, propagation (Radul and Sussman 2009), and models of expected vehicle physics and electromechanical behavior to create causal chains that explain the events leading up to and in an accident (Gilpin and Yuan 2017). This work is also being extended to include commonsense rules of vehicle actions, so that it could monitor planning systems for inconsistent tactics.

Our approach and position is similar to that proposed in Explainable Agency (Langley et al. 2017). This refers to the ability of autonomous agents to explain their decisions and be questioned. Although I adhere to many of the principles of explainable agency, my goal is to extend these principles to *full system design*.

Conclusion and Discussion

Even if machines are robust, their failures are poorly detected and explained. Further, is nearly impossible to inspect if there were any plausible counterfactural decisions. I.e., in the Uber self-driving car case, were there any other planning decisions that could have avoided the trafic

As we add more components to this machines, either to make function autonomously, or to add more capabilities and features, we are also increasing the number of ways that they can fail. With more components and connections, detecting the root-cause becomes difficult, and without a proper reason or detection of error, this can prevent the machine or operator to learn from the failures.

At the current time, a machine can only justify their actions with incomprehensible log trace, unconvincing to those who demand a human-readable justification in order to trust this machines actions. Further, current testing protocols do not accurately mimic real life. Testing in simulation cannot cover all the test cases, how can we ensure that these vehicles are tested properly and how can we ensure that they perform to their best ability in real scenarios?

In"state-of-the-art" diagnostic systems, root cause analysis and human experts are inadequate. Deep neural networks, our most powerful perceptual mechanisms, are opaque to even the most knowledgeable human experts. Even if machines can somehow communicate their failures and anomalies, the appropriate next steps are rarely obvious. For example, a "check engine" light on a vehicle does not indicate a specific failure, but rather indicates a need for unspecified maintenance. Our approach develops the capability for a complex machine to be aware of and report on its internal state, including multiple decisions and failures, supported by reasoning and history.

References

Albanese, M.; Jajodia, S.; Pugliese, A.; and Subrahmanian, V. S. 2011. *Scalable Detection of Cyber Attacks*. Berlin, Heidelberg: Springer Berlin Heidelberg. 9–18.

Chandola, V.; Banerjee, A.; and Kumar, V. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)* 41(3):15.

Garcia-Teodoro, P.; Diaz-Verdejo, J.; Maciá-Fernández, G.; and Vázquez, E. 2009. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security* 28(1-2):18–28.

Gilpin, L. H., and Yuan, B. Z. Getting up to speed on vehicle intelligence. *2017 AAAI Spring Symposium Series*.

Gilpin, L. H., and Yuan, B. Z. 2017. Getting up to speed on vehicle intelligence. In *AAAI Spring Symposium Series*.

Gilpin, L. 2018. Reasonableness monitors. In *The Twenty-Third AAAI/SIGAI Doctoral Consortium at AAAI-18*. New Orleans, LA: AAAI Press.

Grill, M.; Pevnỳ, T.; and Rehak, M. 2017. Reducing false positives of network anomaly detection by local adaptive multivariate smoothing. *Journal of Computer and System Sciences* 83(1):43–57.

Langley, P.; Meadows, B.; Sridharan, M.; and Choi, D. 2017. Explainable agency for intelligent autonomous systems. In *AAAI*, 4762–4764.

Lavin, A., and Ahmad, S. 2015. Evaluating real-time anomaly detection algorithms–the numenta anomaly benchmark. In *Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on*, 38–44. IEEE.

Negash, L.; Kim, S.-H.; and Choi, H.-L. Distributed unknown-input-observers for cyber attack detection and isolation in formation flying uavs.

Radul, A., and Sussman, G. J. 2009. The art of the propagator. In *Proceedings of the 2009 international lisp conference*, 1–10.

Vallis, O.; Hochenbaum, J.; and Kejariwal, A. 2014. A novel technique for long-term anomaly detection in the cloud. In *HotCloud*.