

Chapter 2

Towards Modeling the Behavior of Autonomous Systems and Humans for Trusted Operations

Gavin Taylor, Ranjeev Mittu, Ciara Sibley, and Joseph Coyne

2.1 Introduction

Unmanned systems will perform an increasing number of missions in the future, reducing the risk to humans, while increasing their capabilities. The direction for these systems is clear, as a number of Department of Defense roadmaps call for increasing levels of autonomy to invert the current ratio of multiple operators to a single system (Winnefeld and Kendall 2011). This shift will require a substantial increase in unmanned system autonomy and will transform the operator's role from actively controlling elements of a single platform to supervising multiple complex autonomous systems. This future vision will also require the autonomous system to monitor the human operator's performance and intentions under different tasking and operational contexts, in order to understand how she is influencing the overall mission performance.

Successful collaboration with autonomy will necessitate that humans properly calibrate their trust and reliance on systems. Correctly determining reliability of a system will be critical in this future vision since automation bias, or overreliance on a system, can lead to complacency which in turn can cause errors of omission and commission (Cummings 2004). On the other hand, miscalibrated alert thresholds and criterion response settings can cause frequent alerts and interruptions (high false alarm rates), which can cause humans to lose trust and underutilize a system (i.e., ignore system alerts) (Parasuraman and Riley 1997). Hence, it is imperative that not only does the human have a model of normal system behavior in different

G. Taylor (✉)
United States Naval Academy, Annapolis, MD, USA
e-mail: taylor@usna.edu

R. Mittu • C. Sibley • J. Coyne
Naval Research Laboratory, Washington, DC, USA
e-mail: ranjeev.mittu@nrl.navy.mil; ciara.sibley@nrl.navy.mil; joseph.coyne@nrl.navy.mil

contexts, but that the system has a model of the capabilities and limitations of the human. The autonomy should not only fail transparently so that the human knows when to assist, but autonomy should also predict when the human is likely to fail and be able to provide assistance. The addition of more unmanned assets with multi-mission capabilities will increase operator demands and may challenge the operator's workload just to maintain situation awareness. Autonomy that monitors system (including human) behavior and alerts users to anomalies, however, should decrease the task load on the human and support them in the role of supervisor.

Noninvasive techniques to monitor a supervisor's state and workload (Fong et al. 2011; Sibley et al. 2011) would provide the autonomous systems with information about the user's capabilities and limitations in a given context, which could provide better prescriptions for how to interact with the user. However, many approaches to workload issues have been based on engineering new forms of autonomy assuming that the role of the human will be minimized. For the foreseeable future, however, the human will have at least a supervisory role within the system; rather than minimizing the actions of the human and automating those actions the human can already do well, it would be more efficient to develop a supervisory control paradigm that embraces the human as an agent within the system and leverages on her capabilities and minimizes the impact of her limitations.

In order to best develop techniques for identifying anomalous behaviors associated with the complex human-autonomous system, models of normal behaviors must be developed. For the purpose of this paper, an anomaly is not just a statistical outlier, but rather a deviation that prevents mission goals from being met, dependent on the context. Such system models may be based on, for example, mission outcome measures such as objective measures of successful mission outcomes with the corresponding behaviors of the system. Normalcy models can be used to detect whether events or state variables are anomalous, i.e., probability of a mission outcome measure that does not meet a key performance parameter or other metric.

The anomalous behavior of complex autonomous systems may be composed of internal states and relationships that are defined by platform kinematics, health and status, cyber phenomena and the effects caused by human interaction and control. Once the occurrence and relationships between abnormal behaviors in a given context can be established and predicted, our hypothesis is that the operational bounds of the system can be better understood. This enhanced understanding will provide transparency about the system performance to the user to enable trust to be properly calibrated with the system, making the prescriptions for human interaction that follow to become more relevant and effective during emergency procedures.

A key aspect of using normalcy models for detecting abnormal behaviors is the notion of context; and behaviors should be understood in the context in which they occur. In order to limit the false alarms, effectively integrating context is a critical first step. Normalcy models must be developed for each context of a mission, and used to identify potential deviations to determine whether such deviations are anomalous (i.e., impact mission success). Proper trust calibration would be assisted through the development of technology that provides the user with transparency

about system behavior. This technology will provide the user with information about how the system is likely to behave in different contexts and how the user should best respond.

We present an approach for modeling anomalies in complex system behavior; we do not address modeling human limitations and capabilities in this paper, but recognize that this is equally important in the development of trust in collaborative human-automation systems.

2.2 Understanding the Value of Context

The role of context is not only important when dealing with the behavior of autonomous systems, but also quite important in other areas of command and control. Today's warfighters operate in a highly dynamic world with a high degree of uncertainty, compounded by competing demands. Timely and effective decision making in this environment is challenging. The phrase "too much data—not enough information" is a common complaint in most Naval operational domains. Finding and integrating decision-relevant information (vice simply data) is difficult. Mission and task context is often absent (at least in computable and accessible forms), or sparsely/poorly represented in most information systems. This limitation requires decision makers to mentally reconstruct or infer contextually relevant information through laborious and error-prone internal processes as they attempt to comprehend and act on data. Furthermore, decision makers may need to multi-task among competing and often conflicting mission objectives, further complicating the management of information and decision making.

Clearly, there is a need for advanced mechanisms for the timely extraction and presentation of data that has value and relevance to decisions for a given context. To put the issue of context in perspective, consider that nearly all national defense missions involve Decision Support Systems (DSS)—systems that aim to decrease the cycle time from the gathering of data to operational decisions. However, the proliferation of sensors and large data sets are overwhelming DSSs, as they lack the tools to efficiently process, store, analyze, and retrieve vast amounts of data. Additionally, these systems are relatively immature in helping users recognize and understand important contextual data or cues.

2.3 Context and the Complexity of Anomaly Detection

Understanding anomalous behaviors within the complex human-autonomous system requires an understanding of the context in which the behavior is occurring. Ultimately, when considering complex, autonomous systems comprised of multiple entities, the question is not what is wrong with a single element, but whether that anomaly affects performance of the team and whether it is possible to achieve the

mission goals in spite of that problem. For example, platform instability during high winds may be normal, whereas the same degree of instability during calm winds may be abnormal. Furthermore, what may appear as an explainable deviation may actually be a critical problem if that event causes the system to enter future states that prevent the satisfaction of a given objective function. The key distinction is that in certain settings, it may be appropriate to consider anomalies as those situations that effect outcomes, rather than just statistical outliers. In terms of the team, the question becomes which element should have to address the problem (the human or the autonomy).

The ability to identify and monitor anomalies in the complex human-autonomous system is a challenge, particularly as increasing levels of autonomy increase system complexity and, fundamentally, human interactions inject significant complexity via unpredictability into the overall system. Furthermore, anomaly detection within complex autonomous systems cannot ignore the dependencies between communication networks, kinematic behavior, and platform health and status.

Threats from adversaries, the environment, and even benign intent will need to be detected within the communications infrastructure, in order to understand its impact to the broader platform kinematics, health and status. Possible future scenarios might include cyber threats that take control of a platform in order to conduct malicious activity, which may cause unusual behavior in the other dimensions and corresponding states. The dependency on cyber networks means that a network provides unique and complete insight into mission operations. The existence of passive, active, and adversarial activities creates an ecosystem where “normal” or “abnormal” is dynamic, flexible, and evolving. The intrinsic nature of these activities results in challenges to anomaly detection methods that apply signatures or rules that have a high number of false positives. Furthermore, anomaly detection is difficult in large, multi-dimensional datasets and is affected by the “curse of dimensionality.” Compounding this problem is the fact that human operators have limited time to deal with complex (cause and effect) and/or subtle (“slow and low”) anomalies, while monitoring the information from sensors, and concurrently conducting mission planning tasks. The reality is that in future military environments, fewer operators due to reduced manning may make matters worse, particularly if the system is reliant on the human to resolve all anomalies!

Below we describe research efforts underway in the area of anomaly detection via manifolds and reinforcement learning.

2.3.1 Manifolds for Anomaly Detection

A fundamental challenge in anomaly detection is the need for appropriate metrics to distinguish between normal and abnormal behaviors. This is especially true when one deals with nonlinear dynamic systems where the data generated contains highly nonlinear relationships for which Euclidean metrics aren’t appropriate.

One approach is to employ a nonlinear “space” called a manifold to capture the data, and then use the natural nonlinear metric on the manifold, in particular the Riemannian metric, to define distances among different behaviors.

We view the path of an unmanned system as a continuous trajectory on the manifold and recognize any deviations due to human inputs, environmental impacts, etc. Mathematically, we transform the different data types into a common manifold-valued data so that comparisons can be made with regard to behaviors.

For example, a manifold for an unmanned system could be 12 dimensional, composed of position, pitch, roll, yaw, velocities of the position coordinates, and angular velocities of the pitch, roll and yaw. This 12-dimensional model captures any platform (in fact any moving rigid object’s) trajectories under all possible environment conditions or behaviors. This manifold is the tangent bundle, TM of $SO(3) \times \mathbb{R}^3$. Here $SO(3)$ denotes the set of all possible rotations of the unmanned system which is a Lie group, and $\mathbb{R}^3$ the set of all translations of the platform. Since rotations and translations do not commute, this is not a direct product of $SO(3)$ with $\mathbb{R}^3$. The product between $SO(3)$ and $\mathbb{R}^3$ is a “Semi-Product” $\ltimes$. Non-linear key geometric, dynamical and kinematic characteristics are represented using TM . This manifold model is able to encapsulate the unique structure of the environment, effects of human behaviors, etc. through continuous parameterizations and coherent relationships.

Once we have this manifold model and its Riemannian metric, it is possible to define concepts of geodesic neighborhood and other appropriate measurements and map those to mission cost. Such a mapping is done by designing a weighted cost function with dynamical neighborhoods around a trajectory of the platform. For example, if the weather is good in the morning, the neighborhood is smaller than it would be with bad weather. This innovative manifold method could be used to dynamically identify normal or abnormal behaviors occurring during a mission, taking into consideration whether a mission could be successfully achieved under a given cost constraint. We also have the freedom to adjust normal neighborhoods if a mission suddenly changes while en-route. Our model is robust and captures complicated dynamics of unmanned systems and is able to encapsulate very high dimensional data using only a 12 dimensional configuration space.

The algorithms use continuous parameterizations and coherent relationships and are scalable. Our manifold-based methods provide new techniques to combine qualitative (platform mechanics) and quantitative (measured data) methods and are able to handle large, nonlinear dynamic data sets.

2.4 Reinforcement Learning for Anomaly Detection

The military and commercial communities increasingly rely on autonomous systems to augment their capabilities. For example, power plants feature automatic monitoring and safety features, and the military increasingly employs unmanned systems in denied or politically sensitive theaters. However, it is rare for these

systems to be truly autonomous; generally, fine-grained decisions are made by computer (such as an autopilot), while coarse-grained decisions (such as mission tasking or flight plans) are made by human operator. Thus, autonomy is not a binary on-off switch, but instead lies on a continuum. As technology progresses, it is expected that daily operations will shift along this continuum such that more and more decisions will be made by autonomous machine. This point in the continuum is referred to as supervised autonomy, in which human supervisors are tasked with maintaining awareness and identifying and responding to negative surprises, potentially for multiple systems simultaneously. In this scenario, it is important that accurate measures of mission progress be communicated quickly and succinctly, so that supervisors can direct their attention properly, particularly when overtasked.

In this section we make several points. First, we believe the field of Reinforcement Learning can provide the tools necessary to communicate this type of information. Second, we believe automated feature selection for Reinforcement Learning provides an intuitive way to identify features of interest in autonomous systems. Finally, we believe Reinforcement Learning may provide several additional benefits for negative anomaly detection and human decision-making analysis.

2.4.1 *Reinforcement Learning*

The purpose of this section is to describe the field of Reinforcement Learning, so that the new applications can be better understood. We begin by describing the problem intuitively, before introducing the mathematical background. Imagine, for some new unknown system, a human is presented with telemetry data entirely describing every aspect of the system (position, orientation, velocity, sensor readings, control surface position, etc.) for every second of a large number of missions. Given this data, it is possible to imagine the human learning what the system is capable of doing (for example, when learning about an unmanned aircraft, the human may learn that altitude gain can be achieved by setting particular control surfaces to certain settings, but there is a maximum climb rate which cannot be exceeded). Now imagine an expert assigns a “reward” to each observation; a state in which the mission is completed would receive a high reward, while a state in which the system is damaged would receive a negative reward. A logical question, then, is given some new state, what rewards can we expect to receive in the future? Are there certain actions we can take to maximize this expected future rewards? These questions are very important in a supervised context; if we expect to receive many high rewards, no intervention is required. However, if we are in a state with low expected rewards, retasking may be necessary. Reinforcement Learning provides algorithms for performing this predictive analysis.

Mathematically, the problem is described as a Markov Decision Process (MDP). An MDP is a tuple $M = (\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma)$, where $\mathcal{S}$ is the set of possible states the agent could inhabit, $\mathcal{A}$ is the set of actions the agent can take, $\mathcal{P}$ is a transition kernel describing the probability of transitioning between two states given the

performance of an action (mathematically, $p(s'|s, a)$, where $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$), R is the reward function, and $\gamma \in (0, 1)$ is the discount factor, which is used to describe how much we prefer rewards in the short term to rewards in the long term. The expected future rewards from a given state is defined as the *optimal value* of that state, where the optimal value function is defined by the Bellman equation

$$V^*(s) = R(s) + \gamma \max_a \int_{\mathcal{S}} p(s'|s, a) V^*(s') ds'. \quad (2.1)$$

This function defines the optimal value of a state as the expected, discounted sum of rewards from this state forward. In large or complex domains, it is impossible to calculate V^* exactly, forcing instead the construction of an approximation. A huge number of approaches to perform this approximation exist, each with their own benefits and drawbacks.

Classically, an accurate value function is useful for creating autonomy; if choosing between two states, the decision-making agent should choose the one with higher value. This turns long-term planning into a series of greedy, short-term decisions. Because value function approximation techniques are usually general, and can often be applied to any MDP, advancements in Reinforcement Learning lead to better autonomy in any domain.

This same generality is a drawback in practical application, however. While it is theoretically pleasing to begin with no assumptions or prior knowledge about the agent being observed, this is not usually the case. For example, in the case of a new UAV, the known dynamics of the UAV can usually be used to generate a better autopilot than would be generated by Reinforcement Learning, which ignores this tremendously useful prior information.

Therefore, the remainder of this section focuses not on the traditional applications of a value function for autonomy generation, but instead on new applications of this function in a supervised autonomy context.

2.4.2 Supervised Autonomy

In a supervised autonomy context, we assume the agent under most circumstances is capable of performing its task unassisted; the problem then becomes one of helping a possibly-overtasked human supervisor identify cases which do require human intervention. For example, consider the case of a human tasking multiple UAVs at a time to accomplish a variety of missions. Perhaps one of the systems being watched is making slower-than-expected progress, and so another system will have to be tasked to complete its mission before the opportunity expires. Or, more dramatically, perhaps a malfunction occurs and human intervention is required to land the UAV away from the runway with as little damage as possible, in a location easy for recovery. In these cases, the identification of states from which we expect

to receive low rewards is an appropriate way to identify moments which are unlikely to result in mission success.

Intuitively, the value of a state can be thought of as a one-dimensional measure of predicted success. In scenarios where supervisors must keep situational awareness on several independent agents, the distillation of this information into something quickly digestible and understandable is essential. Because the value function is built based on what is expected to occur, given past behavior and an optimal autopilot, a small or negative value function may be an extremely useful indicator of possible mission failure.

We note that the level of accuracy necessary for providing this feedback to a supervisor or performing this analysis is far less than the level of accuracy an agent requires to actually choose actions based entirely on the value function.

2.4.3 Feature Identification and Selection

This viewpoint also may provide ancillary benefits. Mathematically, many approximation schemes take the form of a linear approximation, where $V(s) \approx \Phi(s)w$, for all $s \in \mathcal{S}$. In this approximation, $\Phi(s)$ is a set of feature values for state s , and w is a vector which linearly weights these features. The quality of the approximation depends in large part on the usefulness of these features. Therefore, techniques which can select the few features from a large dictionary which result in the highest quality approximation can provide a great benefit, and are an active area of study (Kolter and Ng 2009; Johns et al. 2010; Mahadevan and Liu 2012; Liu et al. 2012). Interestingly for this domain, the identification of features which are highly correlated with future success and failure may provide an intuitive means of understanding the domain and in the construction of early-warning alarms.

One such technique for performing automated feature selection while calculating an approximate value function is L_1 -Regularized Approximate Linear Programming (RALP) (Petrik et al. 2010). RALP is unique among other feature-selection approaches in that it has tightly bounded value function approximation error, even when using off-policy samples, and when the domain features a great deal of noise (Taylor and Parr 2012).

RALP works by solving the following convex optimization problem:

$$\begin{aligned} \min_w \quad & \rho^T \Phi w \\ \text{s.t.} \quad & \sigma^r + \gamma \Phi(\sigma^{s'})w \leq \Phi(\sigma^s)w \quad \forall \sigma \in \Sigma \\ & \|w\|_{-1} \leq \psi. \end{aligned} \tag{2.2}$$

Φ is the feature matrix, Σ is the set of all samples, where σ is one such sample, in which the agent started at state σ^s , received reward σ^r , and transitioned to state $\sigma^{s'}$. ρ is a distribution, which we call the state-relevance weights, in keeping with the

(unregularized) Approximate Linear Programming (ALP) terminology of de Farias and Van Roy (2003). $\|w\|_{-1}$ is the L_1 norm of the vector consisting of all weights excepting the one corresponding to the constant feature.

This final constraint, which contributes L_1 regularization, provides several benefits. First, regularization in general ensures the linear program is bounded, and produces a smoother value function. Second, L_1 regularization in particular produces a sparse solution, producing automated feature selection from an over-complete feature set. Finally, the sparsity results in few of the constraints being active, speeding the search for a solution by a linear program solver, particularly if constraint generation is used.

Most relevantly to this problem, the selected features are those most useful in predicting the future receipt of rewards. Practically, these can be interpreted as the few features, from a potentially large candidate set, which carry the strongest signal predicting success or failure; while this definitely improves prediction accuracy, it may also be useful in related tasks in which an intuitive prediction of future autonomous performance will be useful, such as in scenarios of transfer learning.

2.4.4 Approximation Error for Alarming and Analysis

We also propose an exciting direction of research in using the error in approximation as a useful signal in flight analysis and alarming. Consider again the Bellman equation of Eq. (2.1), and a single observation, consisting of a state, a received reward, and the next state (an observation can be denoted (s, r, s') , where $s, s' \in \mathcal{S}$, and $r = R(s)$). Also assume the existence of some approximation $\hat{V}$, such that $\hat{V}(s) \approx V^*(s)$ for all $s \in \mathcal{S}$. If the approximation is exactly correct, and everything goes as expected, then $\hat{V}(s) = r + \gamma \hat{V}(s')$ for all such observations. If this equality does not hold, then we have non-zero *empirical Bellman error*

$$BE(s) = r + \gamma \hat{V}(s') - \hat{V}(s). \quad (2.3)$$

When the Bellman error is non-zero, there are several possible explanations.

The first explanation is that the approximation to the value function was incorrect. This explanation has classically received the most attention, as researchers tried to tune their predictions to be as accurate as possible. However, even if the value function $\hat{V} = V^*$ were calculated exactly, in a non-deterministic environment there would still be non-zero Bellman error.

A second explanation is that the prediction would have been accurate, but the controller (autopilot or human) chose an action which was not optimal. We can expect this to happen consistently to some degree. This is because optimality is defined as choosing the action that maximizes the sum of rewards; however, existing autopilot and human decision making is performed using an approach independent of Reinforcement Learning, and ignorant of the designed reward function. Realistically, all approaches are targeting some approximation of optimal

performance, and it is unreasonable to expect them to precisely agree on optimal decision making. However, we argue that decision making resulting from these differing approaches should not differ too much, and that unexpected large drops in the value function would be interesting in performing objective performance assessment, such as in human training and flight analysis.

A third explanation is that the approximation was accurate, but something unexpected happened during the state transition. A positive example would be if a tailwind pushed the aircraft closer to a target location than would normally be expected; in this case, the value of the next state would be larger than expected. More interestingly for supervised autonomy, however, would be in identifying unexpected drops in value, or negative anomalies. It is perhaps obvious this would be useful in identifying sudden, dramatic drops, as might result from an event such as mechanical failure. More promising, however, is the potential for identifying prolonged periods of underperformance which might indicate a “low-and-slow” anomaly which might otherwise be difficult to detect. These types of anomalies occur when a small error impacts an output over a long period of time. These errors are difficult to detect as they typically do not cross any predefined error boundaries.

To our knowledge, the Bellman error has never been used to perform this type of analysis.

2.4.5 Illustration

To illustrate the insights of Sects. 2.4.2 and 2.4.4, we use data from two different domains. First, we use a simulated domain often used for benchmarking value function approximation algorithms. This domain provides an easily-predicted environment to demonstrate the application of reinforcement learning in an easily-visualized way. Second, we use real world data collected by the in-flight computers of landing TigerShark UAVs to demonstrate the technique’s efficacy even in less-predictable, real-world scenarios.

2.4.5.1 Synthetic Domain

We use a common Reinforcement Learning benchmark domain, in which an autonomous agent learns to ride a bicycle (Randløv and Alstrøm 1998; Lagoudakis and Parr 2003). In this domain, the bicycle begins facing north, and must reach a goal state 1 km to the east. To be precise, a state in the space consists of the bicycle’s angle from perfectly upright, the angular velocity and angular acceleration with respect to the bicycle and the ground, the handlebar angle and angular velocity of the handlebar, and the angle of the goal state from the current direction of travel. Actions include leaning left or right, turning the handlebars left or right, or doing nothing at all. The reward function is a shaping function based on progress made towards the goal.

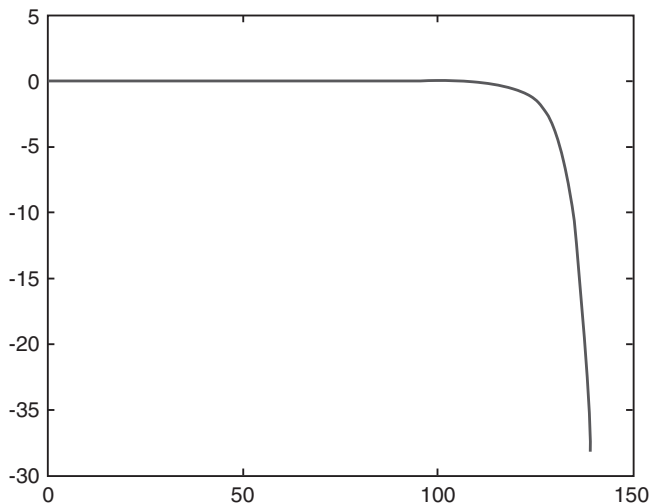


Fig. 2.1 Value as a function of time

Samples to learn from are drawn by beginning in an upright position, and taking random actions until failure. Given a large enough set of such samples, the agent can learn a value function, which it then uses to balance and successfully navigate through the space on a high percentage of trials. To do this, the approximate value function was constructed using RALP, using features composed of a large number of monomials defined on dimensions of the state space.

As our illustration, we consider one run, in which the bicyclist fails to remain upright despite access to a good previously-calculated value function. As the agent moves through the state space, the value function of its current state changes as a function of time. Figure 2.1 graphs this value until failure. Notice that as the bicycle begins to oscillate and eventually collapse, the expectation begins to drop, as it becomes less and less likely the agent will be able to recover. It is easy to imagine an alarm system set off in the beginning stages of this process, alerting a supervisor of the coming failure.

Given the same run, we also plot the absolute value of the empirical Bellman error of Eq.(2.3) as a function of time in Fig.2.2. Notice the predictions were extremely accurate, and Bellman error very low, until shortly before the bicycle began to fall. In analyzing the performance of the agent, this information provides insight into when the initial mistakes were made which would, several time steps in the future, result in a failed mission; in the analysis of the agent's performance, this is crucial information.

If, however, the interest is not in analyzing past performance, but is in identifying anomalies which may require human intervention in real time, the Bellman error illustrated in Fig. 2.2 would again be extremely useful. It is plainly visually obvious when unexpected behavior began, and it is equally clear it correlates with the

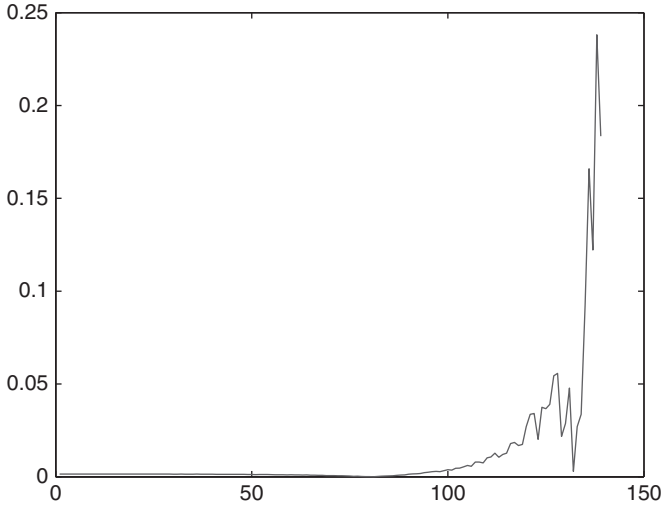


Fig. 2.2 Empirical Bellman error as a function of time

beginnings of the task failure. In particular, we would like to emphasize the anomalies begin to appear well before the bicycle collapses, allowing it to serve as an early warning system.

2.4.5.2 Real-World Domain

The TigerShark Unmanned Aerial Vehicle is a remotely-operated surveillance aircraft with a 22-foot wingspan and weighing 260 pounds, currently in use in theater operations. It can perform autonomous waypoint navigation, or be controlled manually. For this domain, we focus on autonomous alarming for TigerShark landings.

Telemetry data from 1267 landing attempts were collected, for a total of approximately 6.2 million individual samples, which were collected every 1000 ms. Landings were performed both by human pilots and the autopilot. The samples themselves are made up of 192 different sensor readings from the UAV control and sensor surfaces, as well as sensor data taken on the ground. Aside from flight status information like altitude, latitude, longitude, ground speed, etc., the data includes information about the controls coming from the pilot or autopilot, and the connection between the ground station and the UAV.

During landing, pilots have instructions to not allow their roll to extend past a certain angle, to follow a prescribed glide slope, and to keep their aircraft as close to the extended center line of the runway as possible. Therefore, in addition to the sensor readings, features also included the glide slope, the distance from the desired

glide slope squared, the distance from the center line squared, the roll squared, and the reward awarded to the state.

The reward function was a shaping function based around the error from the desired glide slope, the distance from the center line of the runway, and the absolute value of the roll. In addition, the reward function contained additional penalties if the aircraft left the extended center third of the runway, or performed too large of a roll. By rule, either of the latter two cases should result in a decision to abort the landing.

This reward function differs from the one in the bicycle domain in a very significant way. In the synthetic domain, the actions taken by the autonomous agent were taken with the purpose of maximizing the rewards received. In the real-world domain, the actions taken are simply those of a pilot attempting to land an aircraft. As such, we can expect a much larger Bellman error in most states, as the value function is making a prediction on the erroneous assumption the pilot is aware of, and is trying to maximize, the reward function. Nevertheless, we will see the Bellman error is nonetheless helpful.

From the set of all landing samples, 2000 were randomly selected, and a value function computed using RALP. We then applied this value function to two separate flights. In the first flight, though the pilot eventually succeeded in landing the plane, the path was erratic and the landing should have been aborted due to an inability to stay within the allowed center third of the runway, and inability to maintain the roll within allowed parameters.

In Fig. 2.3 we see the value function of this landing. The shaded area is an area of special interest, as it is quite low, and produces the lowest value of the landing. We zoom on this area for both the value function (Fig. 2.4) and the Bellman error (Fig. 2.5). First, we note that both the value function and the Bellman error are

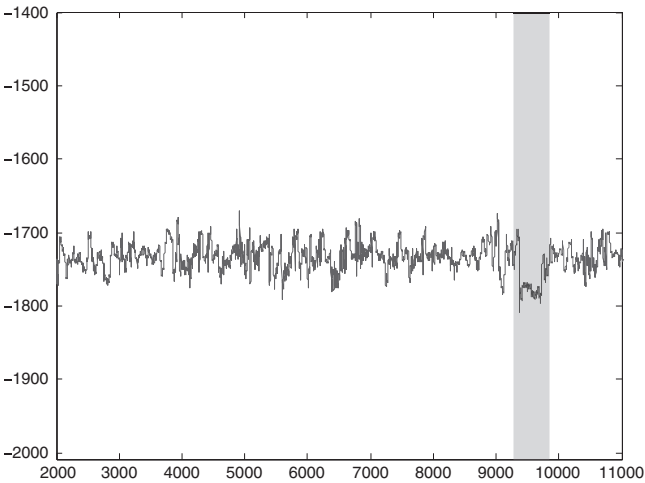


Fig. 2.3 Value as a function of time on unsuccessful flight

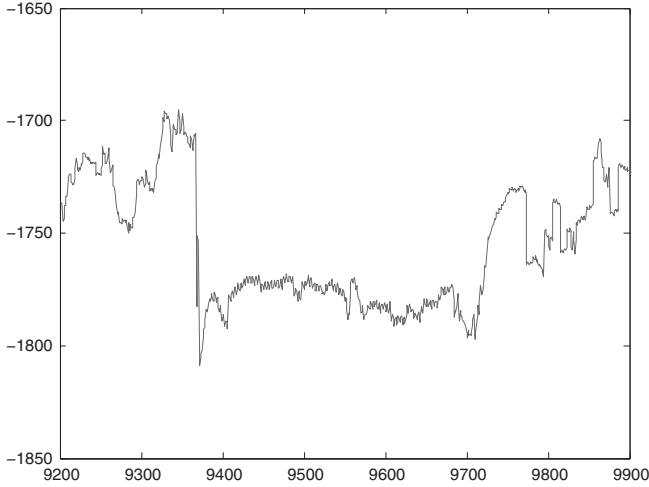


Fig. 2.4 Zoomed value as a function of time on unsuccessful flight

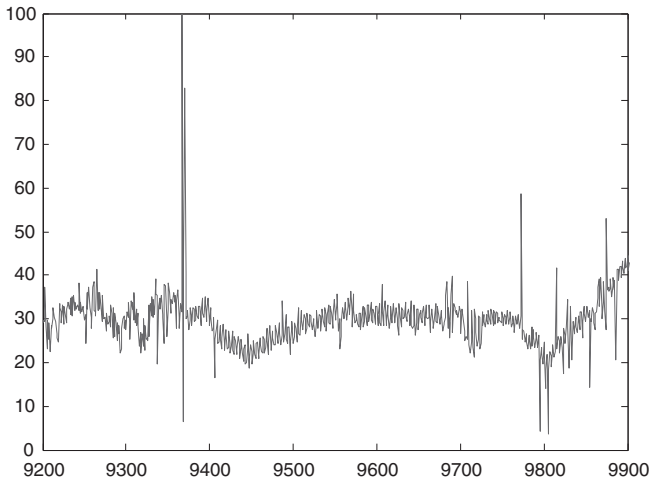


Fig. 2.5 Zoomed empirical Bellman error as a function of time on unsuccessful flight

much less smooth than in the bicycle domain. As discussed in Sect. 2.4.4, this is expected to some degree, as the pilot is not making decisions based on the reward function. Nevertheless, there is still a great deal of information in these two graphs. In particular, we note that the Bellman error spikes greatly in the timestep before the value function begins to drop. This drop results from the aircraft swinging so wide as to cross the boundary for allowed flight, outside the center third of the runway, and trying to recover by banking steeply. At this point, the landing should have been aborted. The spike in Bellman error preceding this drop illustrates the potential for

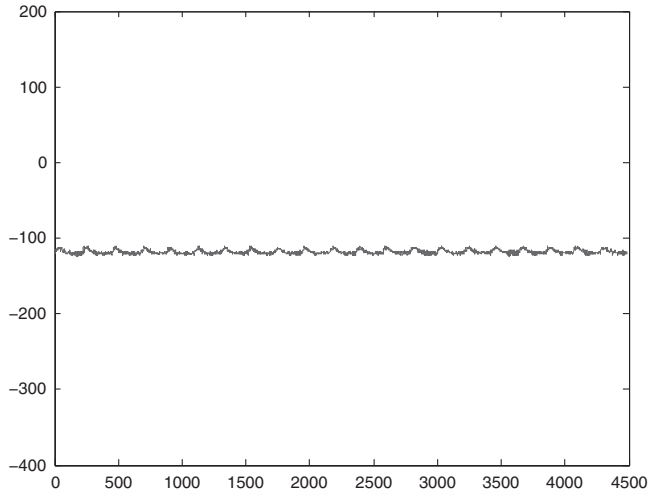


Fig. 2.6 Value as a function of time on successful flight

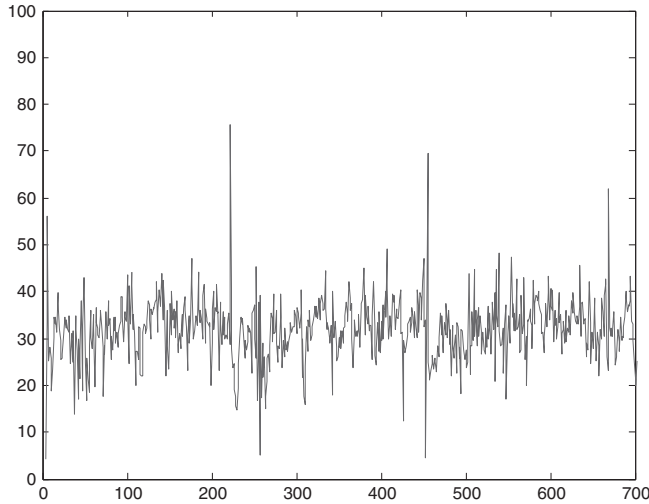


Fig. 2.7 Empirical Bellman error as a function of time on successful flight

a Bellman error-based alarm. In contrast, Figs. 2.6 and 2.7 are the result of a flight without any such negative occurrences (Fig. 2.7 is zoomed to provide the same scale as Fig. 2.5). Note that the two value functions (Figs. 2.3 and 2.6) are scaled identically on the y axis, though the bad value function is much, much lower, as you would expect from a flight struggling to stay on course. Second, we note the value function for the good flight is much less erratic. The Bellman error, meanwhile, remains generally low, with no significant spikes to set off an alarm.

In a scenario of human performance analysis, it is clear from the magnitude of the value function that the pilot for this flight avoided the problems of the first flight; the small magnitude implies the second pilot performed much closer to the way an optimal pilot would be expected to. Additionally, the small oscillations imply a steadier flight.

These illustrations also demonstrate the usefulness of the Bellman error in alarming an autonomous mission; when things went wrong, the Bellman error spiked, providing a concise channel of communication of the error to the operator.

2.5 Predictive and Prescriptive Analytics

It is clear the DoD and the U.S. Navy are increasingly reliant on autonomy, machines and robotics whose behavior is increasing in complexity. Most research indicates operational improvements with autonomy, but autonomy may introduce errors (Manzey et al. 2012) that impact performance. These errors result from many factors, including faulty design assumptions especially in data fusion aids, stochasticity with sensor/observational data, and the quality of the information sources fed into fusion algorithms. Furthermore, additional factors may include greater sophistication and complexity and the subsequent inability of humans to fully comprehend the reasons for decisions made by the automated system (i.e., a lack of transparency). In spite of these known faults, some users rely on autonomy more than is appropriate, known as autonomy “misuse” (Parasuraman and Riley 1997). Another bias associated with human-automation collaboration is disuse, where users underutilize autonomy to the detriment of task performance.

We conjecture that in order to help overcome issues associated with misuse/disuse, the next generation of integrated human-autonomous systems must build upon the descriptive and predictive analytics paradigm of understanding and predicting, with a certain degree of confidence, what the complex autonomous system has done and what it will do next based on what is considered normal for that system in a given context. Once this is achievable, it will enable the development of models that proactively recommend what the user should do in response in order to achieve a prescriptive model for user interactions (Fig. 2.8).

By properly presenting current and future system functioning to the user, and capturing user interactions in response to such states, we believe more effective human-automation collaboration and trust calibration can be established. The key question is “how are the best user interactions captured?”

2.6 Capturing User Interactions and Inference

Transparency in how a system behaves should enable the user to calibrate their level of trust in the system. However, there are still significant challenges that remain

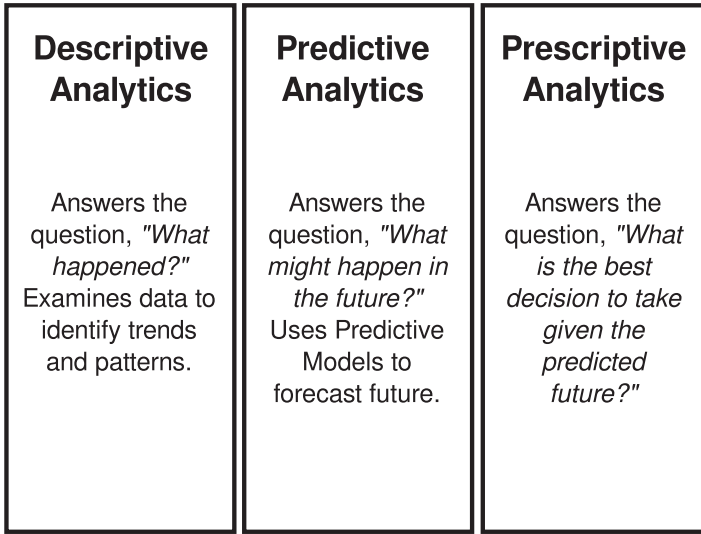


Fig. 2.8 Different forms of analytics

with regard to capturing and understanding the human dimensions of supervisory control in order to provide prescriptions for interaction. We envision several longer term challenges related to the notion of prescriptive analytics, specifically how best to understand and model the information interaction behaviors of the user. These information seeking behaviors may be in reference to the potential anomalies in the system, in relation to what is provided by the on-board sensors, etc. and may require the development of the following capabilities:

- Adequately capturing users' information interaction patterns (and subsequently user information biases)
- Reasoning about information interaction patterns in order to infer decision making context; for example, the work being done by researchers within the Contextualized Attention Metadata community and the Universal Interaction Context Ontology (Rath et al. 2009) might serve as a foundation
- Instantiating formal models of decision making based on information interaction behaviors (potentially using cognitive architectures)
- Leveraging research from the AI community in plan recognition to infer which decision context (model) is active, and which decision model should be active
- Recognizing decision shift based on work that has been done in the Machine Learning community with "concept drift," and assessing how well this approach adapts to noisy data and learns over time
- Incorporating uncertainty and confidence metrics when fusing information and estimating information value in relation to decision utility
- Using models of cognition and decision making (and task performance) to drive behavior development and interface development

Lastly, research is needed to address how the autonomous platform should adapt to user behaviors in order to balance both mission requirements as well as servicing the needs of the human supervisors.

2.7 Challenges and Opportunities

Elaborating on our ideas, longer-term research should be focused on the following: decision models for goal-directed behavior, information extraction and valuation, decision assessment and human systems integration.

With regard to decision models for goal-directed behavior, the key research question may include how to instantiate prescriptive models for decision making, which integrate information recommendation engines that are context-aware. Furthermore, what are the best techniques that can broker across, generalize, or aggregate individual decision models in order to enable application in broader mission contexts? Supporting areas of research may include the development of similarity metrics that enable the selection of the appropriate decision model for a given situation, and intuitive decision model visualizations.

The notion of information extraction and valuation would involve locating, assessing, and enabling, through utility-based exploitation, the integration of high-value information within decision models, particularly in the big data realm. This is a particular research challenge due to heterogeneous data environments when dealing with unmanned systems. In addition, techniques that can effectively stage relevant information along the decision trajectory (while representing, reducing and/or conveying information uncertainty) would enable a wealth of organic data to be maximally harvested.

In reference to decision assessment, research needs to address what are the most effective techniques for modeling decision “normalcy,” in order to identify decision trajectories that might be considered outliers and detrimental to achieving successful outcomes in a given mission context. Furthermore, techniques that proactively induce the correct decision trajectory to achieve mission success are also necessary. Metrics for quantifying decision normalcy in a given context can be used to propose alternate sequences of decisions or induce the exact sequence of decisions. This would require pre-staging the appropriate information needed to support the evaluation of decisions, potentially improving the speed and accuracy of decision making.

Lastly, with regard to human systems integration, the key challenges are in understanding, modeling and integrating the human state (workload, fatigue, experience) as well as the human decision making component as an integral part of the aforementioned areas. Specific topics include: representing human decision-making behavior computationally; accounting for individual differences in ability and preferences; assessing human state and performance in real-time (during a

mission) in order to facilitate adaptive automation; mathematically capturing the human assessment of information value, risk, uncertainty, prioritization, projection and insight; and computationally representing human foresight and intent.

2.8 Summary

The development of robust, resilient, and intelligent systems requires the calibration of trust by humans when working with autonomous platforms. We contend that this can be enabled through a capability which allows system operators to understand anomalous states within the system of systems, which may lead to failures and hence impact system reliability. Likewise, the autonomy should understand the decision making capabilities and other limitations of the humans in order to proactively provide the most relevant information given the user's task or mission context.

This position paper has discussed the need for anomaly detection in complex systems in order to promote a human supervisor's understanding of system reliability. This is a challenging problem due to the increasing sophistication and growing number of sensor feeds in such systems which creates challenges for conducting big data analytics. Technical approaches that enable dimensionality reduction and feature selection should improve anomaly detection capabilities. Furthermore, building models that account for the context of each situation should improve the understanding of what is considered an anomaly. Additionally, we argue that anomalies are more than just statistical outliers, but should also be based upon whether they hinder the ability of the system to achieve some target end state. Understanding anomalies, we believe, should inform, and make more effective, the user's interaction with system. The interaction may include learning more about the anomaly through some form of query, command and control of the situation, entering into some emergency control procedure, etc.

Numerous research questions remain about the most effective interactions between human and autonomy. We believe the following research areas require further exploration in order to build more robust and intelligent systems. First, researchers should seek to capture users' interaction patterns (and subsequently user information biases) and reasoning about interaction patterns in order to infer decision making context. The work being done by researchers within the Contextualized Attention Metadata community and the Universal Interaction Context Ontology (Rath et al. 2009) might serve as a foundation for this approach. Second, instantiating formal models of human decision making based on interaction behaviors would lead to autonomous recognition of human capabilities and habits. Third, leveraging research from the AI community in plan recognition would allow for the inference of active decision contexts (model), and decision model selection. Fourth, adapting work that has been done in the Machine Learning community with concept drift, to recognize decision shifts and assessing how well this approach adapts to

noisy data. Finally, it is necessary to incorporate uncertainty and confidence metrics when fusing information and estimating information value in relation to decision utility.

In order to build trusted systems which include a human component performing supervisory control functions, it is vital to understand the behaviors of the autonomy as well as the human (and his/her interaction with the autonomy). This should provide a holistic approach to building effective collaborative human-automation systems, which can operate with some level of expectation and predictability.

Acknowledgements The collaboration between the Naval Research Lab and the US Naval Academy was supported by the Office of Naval Research, grant number N001613WX20992.

References

- Cummings M (2004) Automation bias in intelligent time critical decision support systems. In: AIAA 3rd intelligent systems conference
- de Farias DP, Van Roy B (2003) The linear programming approach to approximate dynamic programming. *Oper Res* 51(6):850–865
- Fong A, Sibley C, Coyne J, Baldwin C (2011) Method for characterizing and identifying task evoked pupillary responses during varying workload levels. In: Proceedings of the annual meeting of the human factors and ergonomics society
- Johns J, Painter-Wakefield C, Parr R (2010) Linear complementarity for regularized policy evaluation and improvement. In: Lafferty J, Williams CKI, Shawe-Taylor J, Zemel R, Culotta A (eds) *Advances in neural information processing systems*, vol 23, pp 1009–1017
- Kolter JZ, Ng A (2009) Regularization and feature selection in least-squares temporal difference learning. In: Bottou L, Littman M (eds) *Proceedings of the 26th international conference on machine learning*, Omnipress, Montreal, Canada, pp 521–528
- Lagoudakis MG, Parr R (2003) Reinforcement learning as classification: leveraging modern classifiers. In: *Proceedings of the twentieth international conference on machine learning*, vol 20, pp 424–431
- Liu B, Mahadevan S, Liu J (2012) Regularized off-policy TD-learning. In: *Proceedings of the conference on neural information processing systems (NIPS)*
- Mahadevan S, Liu B (2012) Sparse Q-learning with mirror descent. In: *Conference on uncertainty in artificial intelligence*
- Manzey D, Reichenbach J, Onnasch L (2012) Human performance consequences of automated decision aids. *J Cogn Eng Decis Mak* 6:57–87
- Parasuraman R, Riley V (1997) Humans and automation: use, misuse, disuse, abuse. *Hum Factors* 39:230–253
- Petrik M, Taylor G, Parr R, Zilberstein S (2010) Feature selection using regularization in approximate linear programs for Markov decision processes. In: *Proceedings of the 27th international conference on machine learning*
- Randløv J, Alstrøm P (1998) Learning to drive a bicycle using reinforcement learning and shaping. In: *Proceedings of the 15th international conference on machine learning*, pp 463–471
- Rath A, Devaurs D, Lindstaedt S (2009) UICO: an ontology-based user interaction context model for automatic task detection on the computer desktop. In: *CIAO '09: proceedings of the 1st workshop on context, information and ontologies*
- Sibley C, Coyne J, Baldwin C (2011) Pupil dilation as an index of learning. In: *Proceedings of the annual meeting of the human factors and ergonomics society*

- Taylor G, Parr R (2012) Value function approximation in noisy environments using locally smoothed regularized approximate linear programs. In: de Freitas N, Murphy K (eds) Conference on uncertainty in artificial intelligence. Catalina island, California, pp 835–842
- Winnefeld JA, Kendall F (2011) Unmanned systems integrated roadmap FY2011-2036. US Department of Defense. <http://www.acq.osd.mil/sts/docs/Unmanned%20Systems%20Integrated%20Roadmap%20FY2011-2036.pdf>

Robust Intelligence and Trust in Autonomous Systems

Mittu, R.; Sofge, D.A.; Wagner, A.; Lawless, W.F. (Eds.)

2016, XII, 270 p. 89 illus., 64 illus. in color., Hardcover

ISBN: 978-1-4899-7666-6