A Bayesian Decision Theory Paradigm for Test and Evaluation

James P. Ferry
Metron, Inc.
Reston, VA, USA
ferry@metsci.com

Adam S. Ahmed
Metron, Inc.
Reston, VA, USA
ahmed@metsci.com

***Abstract*—Traditional methods for the Test and Evaluation of military systems are based on a combination of ensuring that requirements are met and optimizing the information gained during testing. However, neither approach can address the following fundamental question: how much is a test worth? There are obvious practical benefits to being able to answer this question. Furthermore, the inability of traditional methods to address it suggest that they are not capturing the essence of what the Test and Evaluation process is.**

This paper presents a new approach to Test and Evaluation based on Bayesian Decision Theory. It maintains the current knowledge one has about the parameters that govern a system's behavior, updating this knowledge using a Bayesian filter whenever new data arrive. It couples this Bayesian filter with a utility function that is based on the system's operational utility: i.e., the value of the deployed system to its various stakeholders. In this paradigm, the value of the system to its stakeholders and the cost of testing can be expressed in a common currency of dollars. Therefore it answers the question of how much a test is worth as an organic by-product of how it functions.

This paradigm, called Dynamo, is explored in a simple scenario that demonstrates the insights it offers into the nature of Test and Evaluation. It demonstrates that an arbitrary utility function can be decomposed into an intrinsic component that is concerned only with producing the correct decision, and a cost of imprecision which values information about a system for its own sake. Some testing protocols implicitly optimize utility functions that have a negative cost of imprecision, leading to the pathological behavior of declining to test even when testing is free.

The paper concludes with a discussion of how Dynamo was implemented to perform a case study on test data from a specific radar system.

***Index Terms*—test and evaluation, testing, Bayesian, Bayesian Decision Theory, utility, Moneyball, Dynamo**

I. Introduction

Test & Evaluation (T&E) is essential to the development, procurement, and deployment of military systems. As these systems incorporate a broader range of component algorithms—such as communication protocols, automatic target recognition, and information fusion—their effectiveness and reliability in operational environments become more challenging to ensure. The data fusion community has developed many such component algorithms, enabling capabilities for multi-sensor fusion, target tracking, decision support, and data association. These algorithms are usually grounded in some principled inference framework that captures the essence of the problem to be solved and its relationship to the available data. Such mathematically principled underpinnings equip the fusion community's algorithms with the robustness required to operate in unforeseen operational environments among other, unforeseen components.

This work is funded by DOT&E (Director, Operational Test & Evaluation).

Ironically, the data fusion community's component algorithms are often grounded in a more principled inference framework than the T&E framework used to assess them. Current conceptions of T&E do not capture the essence of the problem it is required to solve. One mindset views T&E as a process to verify that systems meet requirements with a given degree of confidence. Another views T&E through the lens of experimental design, where the focus is on maximizing the information gained from testing. Neither mindset captures the essence of the T&E problem [1]. For example, neither can address the following fundamental question: when is a proposed test of a system worth the test's cost?

The goal of this paper is to present a different conception of T&E, capturing its essence in the mathematically rigorous manner often applied to solving tracking or data fusion problems. This conception is illustrated in Fig. 1. It decomposes T&E into a choice of *modeling framework* and various *evaluation criteria*. The most basic modeling framework is not to have a model. In this case raw data is processed into a collection of input/output pairs: i.e., *environments* $\mathbf{x}$ in which a test is performed and the *outcomes* y of the test. This simple framework can support a corresponding evaluation criterion based on requirements. For example, the raw data may be processed into a simple set of environments $\mathbf{x} \in \{A, B, C\}$ and pass/fail outcomes y, with requirements on the fraction of y that pass in each environment.

The problem with the input/output framework is that the space of operational environmental conditions is much larger and more complex than $\{A, B, C\}$. When a system is used in environment $\mathbf{x}$, one cannot expect a handbook that provides the statistics of system behavior over, say, 100 trials in environments similar to $\mathbf{x}$. Instead, one must somehow predict what values of y are likely in environment $\mathbf{x}$ in a manner consistent with the test data. To do this systematically is to define a *model* of the probability distribution of y for any given $\mathbf{x}$. Such a model is often parameterized by a vector θ that is then learned from the test data. The *likelihood function* of y is written $L(y|\mathbf{x}, \theta)$. There is a rich body of literature on the

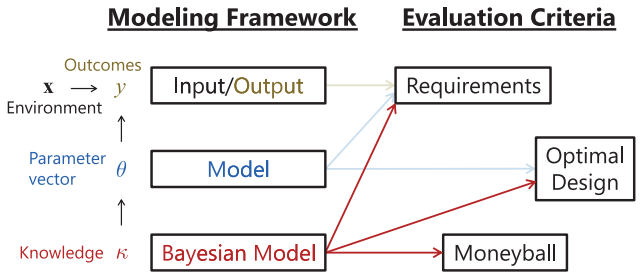

Fig. 1. Bayesian T&E models support Moneyball and other evaluation criteria.

optimal design of a test event to gain information about θ as efficiently as possible [2], [3].

Standard optimal design produces static test plans because it does not account for uncertainty in the estimate of θ during testing. Bayesian inference posits a prior probability distribution on θ and can update this distribution whenever new data arrive [4], [5]. This Bayesian updating is sometimes called sequential Bayesian analysis [6] or, in the context of tracking, a Bayesian filter [7]. The probability distribution on θ may be expressed as $\Pr(\theta|\kappa)$ where κ is a compact representation of all prior knowledge and data accumulated thus far during testing, i.e., a set of hyperparameters. The knowledge κ fully characterizes the system's behavior via the *posterior predictive distribution*,

$$\Pr(y|\mathbf{x},\kappa) = \mathbb{E}_{\theta|\kappa}\left[L(y|\mathbf{x},\theta)\right]. \tag{1}$$

Bayesian reasoning can be used to improve static optimal design [8], [9] or, in conjunction with sequential analysis, to produce Bayesian adaptive designs [10].

Experimental design plays an important role in T&E, and Dynamo could be viewed as providing a form of Bayesian adaptive experimental design. However, Dynamo is grounded in a somewhat different part of the literature: Bayesian Decision Theory (BDT) [6]. BDT was developed in the 1950s by Leonard J. Savage and Dennis Lindley as part of their program to make existing statistical practices mathematically rigorous. Abraham Wald's 1950 book *Statistical Decision Functions* was the key precursor to BDT [11]. Savage had been developing similar ideas under the mentorship of John von Neumann, but considered Wald's book to be foundational [12]. Savage's 1954 book *The Foundations of Statistics* extended Wald's work by developing a Bayesian version of decision theory, emphasizing the importance of subjective probability and expected utility [13]. Lindley developed the sequential version of BDT [14], [15] used in Dynamo. Whereas Lindley's focus was on rigorous underpinnings, Raiffa and Schlaifer developed a popular variant of sequential BDT based on computing the Expected Value of Sample Information (EVSI) for decision trees [16].

BDT requires defining the *utility* of a "state." In Dynamo, this state is κ: the current knowledge about a system under test. BDT also requires specifying the space of possible actions that can be taken. The utility of κ may then be expressed as the maximum, over all possible actions, of the expected utility of the action's result. For example, if the possible actions are selecting which environment $\mathbf{x}$ to test next, then (1) provides the probability distribution over the resulting outcomes y. Bayesian updating then processes κ, $\mathbf{x}$, and any y into the updated knowledge state κ^+, so the utility of the action "test in environment $\mathbf{x}$" can be assessed by taking the expected utility over the resulting states κ^+. This is challenging because the utility of a κ^+ is itself the maximum over all possible actions one could take at that point, and the recursion continues forward indefinitely. When possible, this recursion may be solved exactly using dynamic programming [17]. Section II demonstates how this is done in a simple scenario. In more complex scenarios, it will be necessary to use approximate dynamic programming methods [18] or to develop appropriate decision rules using a Bayesian analog [19] of methods first developed by Wald [20].

In principle, the BDT approach gets at the essence of the T&E problem, but there is a caveat. The utility functions used in practice tend to be motivated by the information mindset [14]. A utility function may be expressed in terms of the entropy of the probability distribution over the parameter vector θ, or it may be combined with the requirements mindset to quantify information about whether requirements are being met. Either way, this can lead to a variety of choices of precisely how to define what information to optimize. Our view is that a different type of utility function is needed: this is labeled "Moneyball" in Fig. 1.

Moneyball is the title of Michael Lewis's book about the development of metrics that assess baseball players by quantifying their value in terms of their contribution to team success, relative to their monetary cost [21]. Moneyball changes the mindset of testing from gathering information or meeting requirements to maximizing value to the stakeholders, whose concerns are encoded in a utility function [1], [22]. We use the term *Dynamo* for the combination of the *Dyna*mic Knowledge κ and the *Mo*neyball utility function. Section II demonstrates the Dynamo paradigm for a simple case.

II. Dynamo for the Beta–Bernoulli Model

A simple type of system to test is one whose outcomes are either $y = 0$ (miss) or $y = 1$ (hit). Ignoring environmental factors, this system can be modeled as a Bernoulli process with a single unknown parameter: the hit probability $\theta = p$. Thus $L(y|p) = p$ for $y = 1$ and $1 - p$ for $y = 0$. One's knowledge κ about p at any time is the set of hyperparameters that fully specify the probability distribution of p. This will initially be informed by subject matter expertise, and it is convenient when this expertise has the form of a beta distribution:

$$\Pr(p|a,b) = \mathrm{B}(p;a,b) \doteq \frac{1}{\mathrm{B}(a,b)} p^{a-1}(1-p)^{b-1}. \tag{2}$$

Thus $\kappa = (a,b)$. The Bayesian update of $\Pr(p|a_0,b_0)$, after observing h hits and m misses, is $\Pr(p|a_0+h,b_0+m)$. The simplicity of this update rule is due to the beta distribution being a *conjugate prior* to the Bernoulli distribution [5], [16].

Consider a test event for such a hit–miss system in which each trial has cost c_T. How long should one continue to test? The information mindset suggests testing until the posterior distribution on p is sufficiently narrow; the requirements mindset suggests testing until p can be resolved to lie above or below some specified threshold with some specified probability. The former emphasizes resolving the *variance* of the parameter distribution; the latter, the *mean*. However, neither mindset is sensitive to the trial cost c_T. This suggests that neither mindset captures the essence of the problem.

The Moneyball mindset is to define the value $u_A(a,b)$ of accepting the system into the next phase of the acquisition cycle. This is challenging in two ways. One is establishing an overall baseline value for, say, a system with known $p = 1$. If the next phase were full-rate production, this would involve an analysis of how many units of the system would be produced, what the manufacturing and sustainment costs would be for each, and what its operational impact would be relative to existing systems. Here this baseline value is set to 1, so c_T is expressed as a fraction of the baseline value. The other challenge is how to specify the functional form of $u_A(a,b)$.

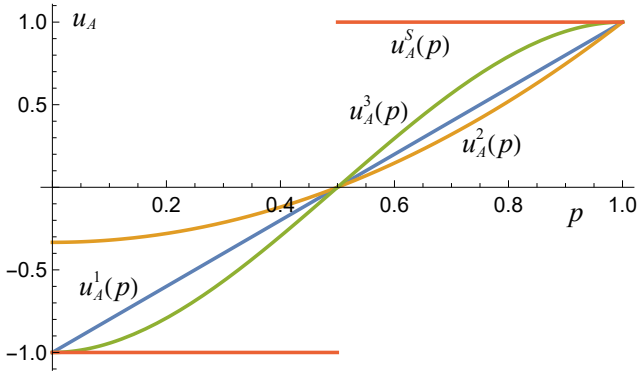

Fig. 2. Four options for acceptance utility as a function of p

One way to specify $u_A(a,b)$ is to begin by specifying $u_A(p)$: the utility of a system for which p is known exactly. Equation (3) gives several options, which are plotted in Fig. 2:

$$\begin{aligned} u_A^1(p) &\doteq 2p-1, & u_A^3(p) &\doteq -4p^3+6p^2-1, \\ u_A^2(p) &\doteq (4p^2-1)/3, & u_A^S(p) &\doteq 2H(p-0.5)-1. \end{aligned} \tag{3}$$

Here $H(x)$ is the Heavyside (or unit step) function.

Three options for $u_A^k(p)$ are degree-k polynomials satisfying $u_A(1) = 1$ and $u_A(0.5) = 0$. This establishes 0.5 as a common crossover point between good and bad systems for comparison. The fourth option, $u_A^S(p)$ is a step function which provides a strict binary valuation of good vs. bad. However one specifies $u_A(p)$ for known p, the next step is to specify $u_A(a,b)$ for unknown p. A natural way to do this is to observe that the system's hit probability is simply the expected value of p. Therefore one could define the following *instantaneous utility*

$$u_A^{k0}(a,b) \doteq u_A^k\left(\mathbb{E}_{p|a,b}[p]\right), \tag{4}$$

indicated by the "0" superscript. A problem with the instantaneous utility is that the expected hit probability $\mathbb{E}_{p|a,b}[p] = a/(a+b)$ keeps changing as the system is used (i.e., as more hits are added to a and misses to b). An alternative is to define the following *intrinsic utility* (with superscript "I"):

$$u_A^{kI}(a,b) \doteq \mathbb{E}_{p|a,b}\left[u_A^k(p)\right]. \tag{5}$$

In this case, p, and hence $u_A^k(p)$, are regarded as intrinsic properties of the system, and the utility of (a,b) is just the expected value of $u_A^k(p)$ over p. Finally, we could also specify a utility function $u_A(a,b)$ directly, without reference to some underlying $u_A(p)$. An example of this is the compliance utility:

$$u_A^C(a,b) \doteq \begin{cases} 1 & \text{if } \Pr(p \geq 0.5|a,b) \geq 0.95, \\ -1 & \text{otherwise.} \end{cases} \tag{6}$$

This utility function assigns a utility of 1 when the system meets the requirement $p \geq 0.5$ with at least 95% probability and assigns utility -1 both to bad and to uncertain systems.

Given the following four quantities,

- The acceptance utility function $u_A(a,b)$,
- The cost c_T of each trial,
- The parameters (a_0, b_0) of the prior on p, and
- The maximum number of trials permitted n_{max},

one can compute a *decision chart* that provides the optimal test decision at any point during test. For any pair (h,m) of the numbers of hits and misses observed thus far, the decision chart specifies whether to continue testing, or to stop and accept or reject the system.

Producing the decision chart begins with computing a *utility chart* containing the *optimal utilities* $u(a,b)$: i.e., the utility of the optimal action in each state (a,b). The possible actions are Accept, Reject, and Continue testing. The first two are each an action to terminate followed by a decision about the system. In general, let D denote the set of such *terminal decisions*, with $D = \{A, R\}$ in this case. One must define a utility $u_d(a,b)$ for each $d \in D$. These are the boundary conditions between the test event and the next phase of the acquisition cycle. The options for $u_A(a,b)$ are discussed above. A rejected system provides no value, so $u_R(a,b) = 0$. The optimal terminal utility is then defined as

$$u_D(a,b) \doteq \max_{d \in D} u_d(a,b) = \max\left(u_A(a,b), 0\right). \tag{7}$$

The optimal utility $u(a,b)$ equals $u_D(a,b)$ whenever there is no option to continue testing.

Otherwise $u(a,b)$ is computed via Bellman recursion [17]. First, the utility of the Continue testing action is backpropagated from larger values of $a+b$:

$$u_C(a,b) \doteq \frac{a}{a+b}u(a+1,b) + \frac{b}{a+b}u(a,b+1) - c_T, \tag{8}$$

where $a/(a+b)$ is the hit probability in state (a,b). The optimal utility $u(a,b)$ is the maximum of the terminal and continue actions:

$$u(a,b) \doteq \max\left(u_D(a,b), u_C(a,b)\right). \tag{9}$$

Once the optimal utility is computed for all a and b over a range of interest, the corresponding optimal decision $e(a,b)$

may be gleaned from whether the Accept, Reject, or Continue option had the greatest utility.

Fig. 3 shows the utility $u^{1I}(a,b)$ and corresponding decision $e^{1I}(a,b)$ for all $a+b \leq 200$ for the intrinsic acceptance utility $u_A^{1I}(a,b)$ derived from $u_A^1(p) \doteq 1-2p$ using (5). The testing cost is $c_T = 10^{-5}$ per trial, the hyperparameters (a_0, b_0) of the prior on p are discussed below, and the maximum number of trials permitted is effectively infinite ($n_{max} = 20,000$). The utility chart itself is not very interesting: the way it drives decisions is only manifest in subtle variations not visible in the figure. The decision chart is more informative. It shows when to Accept, Reject, or Continue testing in green, red, and gray, respectively.

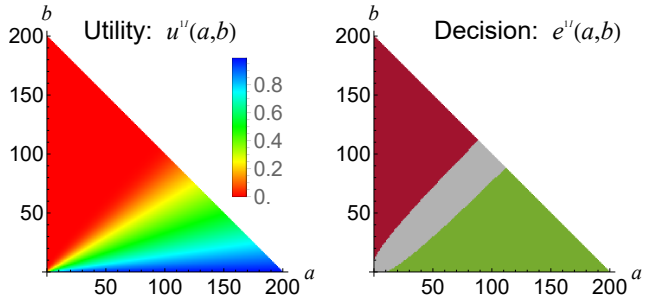

Fig. 3. Utility and decision charts for intrinsic, linear case. **Green** = Accept, **Red** = Reject, and **Gray** = Continue testing

For example, suppose subject matter expertise suggests that the prior distribution on p is $\mathrm{B}(p; a_0, b_0)$ with $(a_0, b_0) = (6, 4)$. This is a distribution with mean 0.60, standard deviation 0.15, and a 25% chance that $p < 0.5$ for the system under test. The point $(6, 4)$ in the decision chart is gray, meaning Continue testing. Now suppose that a data stream of hits and misses arrives and, as (a, b) is updated, $e^{1I}(a, b)$ remains gray until, after $h = 20$ and $m = 6$ misses, $e^{1I}(a, b)$ first becomes green when $(a, b) = (a_0 + h, b_0 + m) = (26, 10)$. At this point, the distribution on p is $\mathrm{B}(p; 26, 10)$ which has mean 0.72, standard deviation 0.07, and a 0.3% chance that $p < 0.5$. A total of 0.00026 has been spent on testing so far. Even though testing is quite cheap, the small chance that further testing could reveal Reject to be a better decision is not worth the expense of testing.

It is essential to emphasize the nature of the result shown in Fig. 3. It should not be thought of as an example of a solution approach to be kept in one's toolbox for similar problems, compared with various machine learning algorithms, etc. Instead, the purpose is to demonstrate what is involved in

- Formulating a Bayesian system model, and
- Making explicit assumptions about stakeholder priorities.

Fig. 3 illustrates the exact results of these explicit choices. Unlike with the requirements and information mindsets, the cost c_T is a necessary part of the formulation. Naturally, when c_T is larger, the gray Continue testing region grows smaller. This is demonstrated in [22] for examples with finite n_{max}, where the gray regions "pinch off" on the terminal diagonal.

Decision charts are, in particular, a result of the explicit assumptions made about the form of the acceptance utility $u_A(a, b)$. As such, these charts provide a diagonostic for the logical consequences of these assumptions. Fig. 4 shows the decision charts for the other intrinsic acceptance utilities $u_A^{kI}(a, b)$ and for the compliance case $u_A^c(a, b)$ in (6). The quadratic and cubic cases have gray Continue regions slightly narrower and wider than in the linear case, respectively, in accordance with the corresponding slopes of $u_A^k(p)$ at $p = 0.5$ in Fig. 2. The gray regions for the step and compliance cases are wider than these, in accordance with their drastic, binary reward structure.

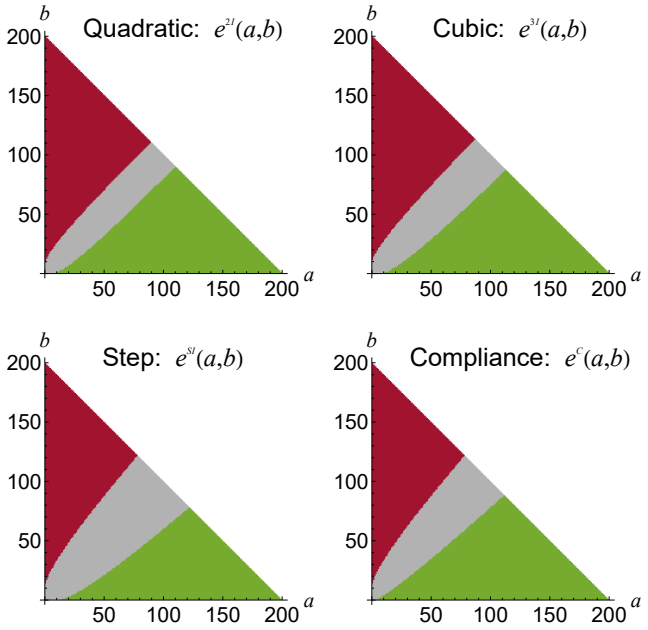

Fig. 4. Decision charts for the intrinsic utility using quadratic, cubic, and step functions, and for the directly defined compliance case.

The intrinsic utility construction (5) yields decision charts with simply shaped decision regions. Using the instantaneous utility construction (4) can yield more complex results. Fig. 5 shows decision charts derived from $u_A^2(p)$ and $u_A^S(p)$ using (4). The gray Continue region in the quadratic case has a second lobe that bulges out from the primary one. This is due to the structure of its utility function: it will be shown below that this is because it rewards knowledge for its own sake. The step function case, on the other hand, has a green Accept region that absorbs any state (a, b) with $a > b$ because the utility achieves the maximal possible value of 1 there, so there can be no point to further testing. In particular, this behavior extends all the way to the corner $(1, 1)$, unlike the decision charts in Fig. 4, which do not accept systems with a probability distribution as broad as $\mathrm{B}(p; 2, 1)$.

In the linear case, decision charts for the intrinsic and instantaneous constructions are identical. Fig. 6 shows the decision chart for the instantaneous, cubic case, which looks "normal" compared to those in Fig. 5. However, something problematic is happening even in this case. We may define

the *value* of the Continue option for any point (a, b) as the maximum price one would be willing to pay for it. The Accept or Reject decision is optimal when this value is below c_T. In general, the value of a trial is large when it is near the $a = b$ boundary because this is where it is most ambiguous whether $p > 0.5$. Fig. 6 exhibits this behavior for the instantaneous, cubic case. However, it also has a region in which the value of a trial is negative: i.e., even if a test were offered for free, the prospect of this additional information is deemed undesirable. This is, in fact, the correct behavior for this utility function, but it is a problematic attitude to have toward truth.

III. General Properties

Despite its simplicity, the beta–Bernoulli model of Section II reveals important properties of the T&E process. These are best explained in a more general context. The graphical model in Fig. 7 depicts a more general scenario: in particular, one which includes the ability to test in different environments $\mathbf{x}$. Here we posit an underlying *adapted stochastic process* with instances of the form $(\theta, \mathbf{x}_1, y_1, \mathbf{x}_2, y_2, \dots)$ and conditional probabilities

$$\begin{aligned} \Pr(\mathbf{x}_n|\theta, \mathbf{x}_1, \dots, y_{n-1}) &= \Pr(\mathbf{x}_n|\mathbf{x}_1, \dots, y_{n-1}), \text{ and} \\ \Pr(y_n|\theta, \mathbf{x}_1, \dots, \mathbf{x}_n) &= L(y_n|\mathbf{x}_n, \theta). \end{aligned} \tag{10}$$

Equation (10) specifies an arbitrary, stochastic rule for generating the next test environment $\mathbf{x}_n$, using all previous test environments and results (but not θ), whereas y_n is assumed to depend only on $\mathbf{x}_n$ and θ. The knowledge κ_n is defined implicitly as

$$\Pr(\theta|\kappa_n) \doteq \Pr(\theta|\mathbf{x}_1, \dots, y_n). \tag{11}$$

Thus κ_0 encodes the prior $\Pr(\theta|\kappa_0) = \Pr(\theta)$, whereas κ_n represents the knowledge implicit in κ_0 and the first n trials. The specific representation of κ_n could be κ_0 and the first n trials themselves or some sufficient statistic derived from them (such as $\kappa = (a, b)$ in the beta–Bernoulli case).

The space D of terminal decisions for a test event may include options to accept the system into the next phase of testing, to send it back to a manufacturer to address a specific issue, or to reject it entirely. Each terminal decision $d \in D$ has a corresponding terminal utility function $u_d(\kappa)$ that is

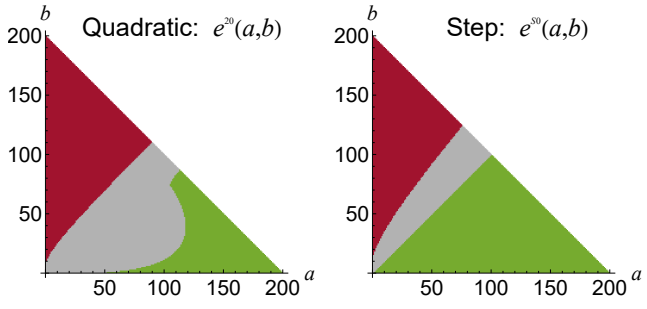

Fig. 5. Decision charts for the instantaneous construction: quadratic and step function cases.

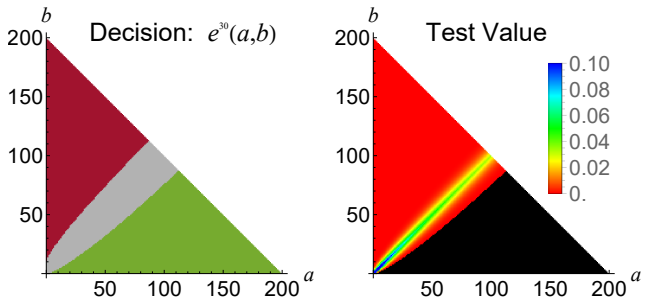

Fig. 6. Decision chart for the instantaneous, cubic case, with the corresponding value provided by a single trial. **Black** = Negative value.

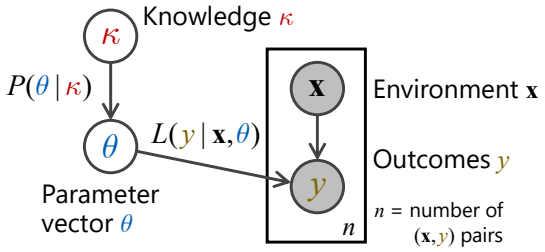

Fig. 7. The knowledge κ encodes the probability distribution of the parameter vector θ. The parameter vector θ and the environment $\mathbf{x}_i$ govern the distribution of the outcome y_i for trials $i = 1$ through n of a test event.

formulated according to what happens next. A decision A to accept the system into full rate production will have a different type of utility function than, say, a decision Q to return the system to its developer to fix a specific subsystem. In general, there will be a boundary in κ-space where $u_A(\kappa)$ and $u_Q(\kappa)$ are tied for the best terminal decision. As indicated by Fig. 6 testing near this boundary tends to be valuable because it helps resolve the correct terminal decision.

Section II indicates that different ways of defining $u_d(\kappa)$ can lead to complexities or pathologies. Fortunately, this is simple to analyze. Any terminal utility function $u_d(\kappa)$ over κ-space determines a corresponding $u_d(\theta)$ (where the notation u_d is now overloaded) by specializing to the case in which κ indicates precise knowledge of θ. On the other hand, given any $u_d(\theta)$, the intrinsic utility construction yields

$$u_d^I(\kappa) \doteq \mathbb{E}_{\theta|\kappa}\left[u_d(\theta)\right]. \tag{12}$$

Any $u_d(\kappa)$ determines an intrinsic utility function $u_d^I(\kappa)$ via this two-step process. The discrepancy between the two is

$$c_d(\kappa) \doteq u_d^I(\kappa) - u_d(\kappa). \tag{13}$$

We call $c_d(\kappa)$ the *cost of imprecision*. It represents how much u_d would increase, on average, were the value of θ known.

When $c_d(\kappa) = 0$, the utility $u_d(\kappa)$ is already the intrinsic utility. Decision charts for these cases are shown in Fig. 3 and in three panels of Fig. 4. These charts have a simple structure and do not conceal any problematic behavior. The instantaneous construction introduces complications. When $u_d(\theta)$ is convex, Jensen's inequality ensures that the cost of

imprecision $c_d(\kappa) \geq 0$. A positive $c_d(\kappa)$ seems desirable, but the decision chart for the instantaneous quadratic case in Fig. 5 has a complex structure. The secondary lobe bulging out exists because of the incentive provided by $c_A^{20}(a,b)$. The acceptance utility functions $u_A^C(a,b)$, $u_A^{S0}(a,b)$, and $u_A^{30}(a,b)$ are more problematic, however, because $c_A(a,b)$ can be negative. For values of (a,b) with $c_A(a,b) < 0$, even a trial offered for free would be considered undesirable because it would decrease u_A on average. When a utility function $u_d(\kappa)$ has a cost of imprecision $c_d(\kappa)$ that is ever negative, we label it *alethophobic* (i.e., "truth-fearing"). This alethophobic behavior can be observed in real test events whenever a tester "checks a box" for some aspect of compliance and is careful not to perform further trials lest the box become unchecked.

Rewriting (13) as $u_d(\kappa) = u_d^I(\kappa) - c_d(\kappa)$ suggests a simple, general method for constructing $u_d(\kappa)$: define

- The utility $u_d(\theta)$ of the parameter vector θ, and
- The cost $c_d(\kappa) \geq 0$ of imprecision (if any).

Explicitly constructing $u_d(\kappa)$ in this way allows one to formulate utility over θ-space, which is much simpler than κ-space. It explicitly formulates the cost $c_d(\kappa)$ rather than letting it arise as the by-product of some other definition, allowing the modeler to articulate what additional value there is in knowing κ precisely. The simple requirement that $c_d(\kappa) \geq 0$ eliminates the problematic alethophobic behavior.

The cost of imprecision $c_d(\kappa)$ connects Dynamo with standard approaches to Bayesian experimental design, which tend to use information-theoretic metrics to penalize imprecision in the same way. What is particularly interesting, however, is that the results of Section II indicate that this is a secondary effect. Setting $c_d(\kappa) = 0$ produces simple decision charts, and this appears to be the dominant mechanism shaping these charts. This makes sense from a decision-theoretic perspective: the main goal of T&E, in this view, is making the correct terminal decisions rather than gaining information for its own sake.

Understanding the simple form of decision charts when $c_d(\kappa) = 0$ begins with the following property of intrinsic utility functions:

$$u_d^I(\kappa) = \mathbb{E}_{\kappa^+|\kappa,\mathbf{x}}\left[u_d^I(\kappa^+)\right]. \tag{14}$$

The expectation in this case is over the updated knowledge κ^+ that arises from the current knowledge κ after a trial in environment $\mathbf{x}$ is performed. The meaning of (14) is that any environment $\mathbf{x}$ that one tests in will, on average, yield the same future utility as the current utility $u_d^I(\kappa)$. Another way of looking at this is to define a sequence of random variables $U_{d0}^I, U_{d1}^I, U_{d2}^I, \ldots$ corresponding to the utility after n tests for all $n \geq 0$. This requires a sequence of test environments $\mathbf{x}_n$ whose evolution law is given in (10). Equation (14) shows that this sequence of random variables forms a *martingale* [23].

Martingales were developed to prove that one cannot beat a fair game with some clever betting strategy. This raises the question of why to test at all if the expected utility remains constant. If the space of terminal decisions D had only one option d, then this would, in fact, be the intuitively correct behavior. That is, if the terminal decision d were preordained, and $c_d(\kappa) = 0$ (i.e., there is no incentive to reduce imprecision for its own sake), then there would indeed be no point to testing. The mechanism behind Dynamo's impetus to test only arises when there are at least two terminal decisions in D. Although each individual $\{U_{dn}^I\}_{n=0}^{\infty}$ is a martingale, the random variables

$$U_{Dn}^I \doteq \max_{d\in D}\left(U_{dn}^I\right) \tag{15}$$

form a *submartingale*: i.e., a sequence of random variables whose current value is always less than or equal to the expected future value. This may be illustrated by a Gaussian random variable Y with mean 1 and standard deviation 3. The expected value of Y is 1, but the expected value of $\max(Y, 0)$ is larger (1.76) because the negative instances of Y dragging down the mean are mitigated to 0.

There is a simple geometry driving the impetus to test for intrinsic utility functions. It arises from partitioning κ-space into *terminal decisions regions* based on which $d \in D$ has maximal $u_d^I(\kappa)$. These are like the regions of a decision chart, but without the Continue option. The impetus to test is very small when all the projected future values of κ^+ (after, say, n more tests) lie in the same terminal decision region. In this case the terminal decision is nearly a foregone conclusion, so there is little reason to test. Testing is valuable, however, when the terminal decision that will be made is still unclear.

A computational challenge arises when κ-space is too large to compute decision charts exhaustively, as in Section II. One could use forward propagation in this case, but when c_T is small it can be worthwhile to conduct many more trials so that a future κ^+ reaps the benefit of a probability distribution extending over multiple terminal decision regions. The goal is to understand the mechanism that generates the impetus to test for intrinsic utility functions in order to develop good approximation methods to resolve these difficulties in more realistic settings beyond the beta–Bernoulli case.

IV. Bayesian Linear Regression

The beta–Bernoulli model elucidates the geometry and phenomena of optimal testing decisions, but its scope is limited because the outcomes y do not depend on an environment $\mathbf{x}$. To extend the analysis of T&E to include environment variables, consider the Gaussian measurement likelihood function,

$$L(y|\mathbf{x},\theta) = \mathcal{N}(y; \mathbf{x}\mathbf{c}^T, \sigma^2), \tag{16}$$

which is used in classical experimental design [2], [3]. The parameter vector is $\theta = (\mathbf{c}, \sigma^2)$, where $\mathbf{c}$ is a coefficient vector, and σ^2 is the aleatoric uncertainty. By convention, $\mathbf{x}$, $\mathbf{c}$ and other boldface quantities are row vectors of length d, with $x_d = 1$ so that c_d is an intercept coefficient. As in the beta–Bernoulli case, it is convenient to choose $\Pr(\theta|\kappa)$ to be a conjugate prior to the normal distribution $L(y|\mathbf{x},\theta)$ in order to make Bayesian updating simple. In this case, there

are parameters $\mathbf{c}$ to model the mean of y and σ^2 to model its variance, so the conjugate prior has two components:

$$\begin{aligned} \Pr(\theta|\kappa) &= \Pr(\mathbf{c}|\sigma^2, \boldsymbol{\mu}, V)\Pr(\sigma^2|\alpha, \beta), \text{ where} \\ \Pr(\mathbf{c}|\sigma^2, \boldsymbol{\mu}, V) &= \mathcal{N}(\mathbf{c}; \boldsymbol{\mu}, \sigma^2 V), \text{ and} \\ \Pr(\sigma^2|\alpha, \beta) &= \mathcal{IG}(\sigma^2; \alpha, \beta). \end{aligned} \tag{17}$$

This is called the Normal-Inverse-Gamma distribution on $\theta = (\mathbf{c}, \sigma^2)$, and the corresponding knowledge κ has four components: $\kappa = (\boldsymbol{\mu}, V, \alpha, \beta)$. This Normal–Normal-Inverse-Gamma (NNIG) model for y and θ is a standard way to perform Bayesian linear regression.

The Bayesian update rule is usually expressed for a data matrix X whose n rows are the individual environments tested. The likelihood function (16) may be expressed as $L(\vec{y}|X, \theta) = \mathcal{N}(\vec{y}; X\mathbf{c}^T, \sigma^2 I)$ in this case, where $\vec{y}$ is the column vector of outcomes of the n trials. The Bayesian update rule from κ to κ^+ may be expressed as

$$\begin{aligned} V^+ &= (V^{-1} + X^T X)^{-1}, \\ \boldsymbol{\mu}^+ &= (\boldsymbol{\mu} V^{-1} + \vec{y}^T X)V^+, \\ \alpha^+ &= \alpha + n, \text{ and} \\ \beta^+ &= \beta + (\vec{y} - X\boldsymbol{\mu}^T)^T (I + XVX^T)^{-1}(\vec{y} - X\boldsymbol{\mu}^T). \end{aligned} \tag{18}$$

An alternative parametrization of κ expresses the update equations more succinctly. Let $\kappa = (\boldsymbol{\nu}, W, \alpha, \gamma)$ where $W \doteq V^{-1}$, $\boldsymbol{\nu} \doteq \boldsymbol{\mu} W$, and $\gamma \doteq \beta + \boldsymbol{\mu} \cdot \boldsymbol{\nu}$. Then the update rule may be written

$$\begin{aligned} W^+ &= W + X^T X, \\ \boldsymbol{\nu}^+ &= \boldsymbol{\nu} + \vec{y}^T X, \\ \alpha^+ &= \alpha + n, \text{ and} \\ \gamma^+ &= \gamma + \vec{y}^T \vec{y}. \end{aligned} \tag{19}$$

The update rule (19) is more efficient, and it illustrates that the process of Bayesian updating for a conjugate prior family can be expressed as the straightforward accumulation of sufficient statistics [16].

The posterior predictive distribution (1) can also be expressed exactly by marginalizing over all parameter vectors θ. It is a multivariate t-distribution with α degrees of freedom:

$$\Pr(\vec{y}|X, \kappa) = \mathcal{T}_\alpha \left(\vec{y}; X\boldsymbol{\mu}^T, \frac{\beta}{\alpha}(I + XVX^T) \right). \tag{20}$$

The above equations encode the "dynamic knowledge" component of Dynamo for the NNIG linear regression model.

The plan for developing Dynamo into a practical tool has two thrusts. The first is to build a core set of models and understand the general nature of how utility propagation generates decision charts for them. The general utility recursion equations are

$$\begin{aligned} u_X(\kappa) &\doteq \mathbb{E}_{\kappa^+|\kappa,X}\left[u(\kappa^+)\right] - c_T(X), \\ u_D(\kappa) &\doteq \max_{d \in D} u_d(\kappa), \\ u_C(\kappa) &\doteq \max_{X \in \mathcal{X}} u_X(\kappa), \text{ and} \\ u(\kappa) &= \max\left(u_D(\kappa), u_C(\kappa)\right). \end{aligned} \tag{21}$$

The first equation in (21) defines the utility of choosing to test the n environments in the data matrix X. The cost of this test now depends on the nature of the environments in X. The third equation takes the maximum over all continue actions in some action space $\mathcal{X}$, and $u(\kappa)$ is then defined as the maximum utility over all terminal actions D and continue actions $\mathcal{X}$. Although (21) is too complex to solve in full generality, understanding its behavior in simplified scenarios will be the basis for developing principled approximation methods.

V. Application to the AN/TPQ-53 System

The second thrust for Dynamo is to develop implementations for specific systems under test. Dynamo has been implemented to assess test plans for the AN/TPQ-53 counter-battery radar system. This section provides a sanitized sketch of how to put Dynamo into practice for testing a real system.

The analysis began with examining the system's Test & Evaluation Master Plan (TEMP) to understand stakeholder priorities via the standards the system is required to meet. A key requirement for operational effectiveness involves the Point of Origin (POO) error in the system's localization of a munitions fire. Different thresholds are specified for different munition types (called Types 1, 2, and 3 here) and radar operating modes (called Modes A, B, and C).

Based on this, an NNIG Bayesian linear regression model was established, involving several continuum test factors that can be varied during testing, as well as a one-hot encoding of several discrete variables. For each trial, these were collected in an environment vector $\mathbf{x}$ (with a 1 appended, as discussed in Section IV). The outcome y for each trial was chosen to be the log of the POO error. A prior value of κ was chosen to represent the state of knowledge about the system at the beginning of the test event.

An intrinsic utility function was chosen for the system based on the requirements specified for each munition type and operating mode. These were softened versions of the hard thresholds specified in the TEMP. A non-zero cost of imprecision was then imposed. This provided a kind of regularization that improved performance, compensating for the recursion method's inability to sample future κ^+ states thoroughly enough. Finally, testing costs were estimated from the costs of materiel, labor hours, and equipment usage.

Next, data were obtained from a Q-53 test event. After various forms of data cleansing, these were mapped into a sequence of $(\mathbf{x}, y)$ pairs. An initial test of Dynamo was conducted where the space $\mathcal{X}$ of testing options was limited to the actual trials that were performed each day. Dynamo then computed the utility of continuing with the actual test plan vs. the utility of stopping and accepting or rejecting the system. A second test of Dynamo gave it a small set of testing plans to choose from, along with the choices to stop and accept or reject. In each case, Dynamo provided a capability that traditional testing protocols cannot: a principled criterion for stopping a test early. In particular, this criterion is sensitive to the cost of testing.

In addition to outputting the utilities of the various testing options, Dynamo visualizes the state of knowledge κ of a test event as data arrive. Fig. 8 is a panel from Dynamo's visualization of an unclassified proxy of the AN/TPQ-53 test event. This panel shows the state of knowledge about the system's required Operational Effectiveness (OE) based on the probability distribution of the POO error. The right side of the panel shows a *Hinton diagram* for the system's OE in nine cases. The color indicates the mean of Dynamo's assessment of the system's OE, and the size of the square indicates the certainty of this assessment. The left panel shows a roll-up of the nine cases into a single graphic, with the needle indicating mean OE and the shaded wedge an 80% containment region.

VI. Summary

Test & Evaluation is a difficult process, and it will only become more difficult as defense systems grow more autonomous, distributed, and complex. Dynamo is a paradigm that challenges the basic mindset of T&E, arguing that T&E is not fundamentally about gaining information or ensuring that requirements are met, but rather about maximizing the net benefit that testing provides.

This paper highlights both the insights that the Dynamo paradigm provides and the challenges inherent to implementing it. One benefit of Dynamo is that it maintains the full state of knowledge about the system at all times, allowing it to be visualized in various ways through an appropriate user interface. A corresponding challenge is that exact, closed-form updating is available only for a limited family of models – those that have a conjugate prior structure. More general knowledge updating will require Markov chain Monte Carlo methods to maintain the knowledge κ [5].

The Moneyball mindset for defining operational utility is compelling: clearly one should conduct the tests that have the greatest net benefit. A challenge involved is that testing cost is fairly straightforward to convert into the common currency of dollars, but the operational benefit of a system is not. Nevertheless, any rational system for determining a fair price for a test must make this conversion. Dynamo does it explicitly by computing the utility associated with the knowledge κ in the same units as the cost of testing.

Formulating the utility of dynamically evolving knowledge states yields important insights into the appropriate structure for a utility function. In particular, any utility function $u_d(\kappa)$ for a terminal decision d can be decomposed into an intrinsic utility function $u_d^I(\kappa)$ and a cost of imprecision $c_d(\kappa)$. This cost $c_d(\kappa)$ governs two important phenomena. When it is always positive it provides an immediate incentive to test in order to reduce uncertainty. When it is ever negative, it produces alethophobia, which inappropriately disincentivizes further testing even when it is free. The intrinsic utility $u_d^I(\kappa)$ provides the impetus to test in order to make the correct decisions. It arises at boundaries between decision regions and backpropagates to earlier κ states in a manner that is geometrically simple but hard to compute in practice. A deeper understanding of how the impetus to test arises from these decision boundaries will help T&E address its current and future challenges.

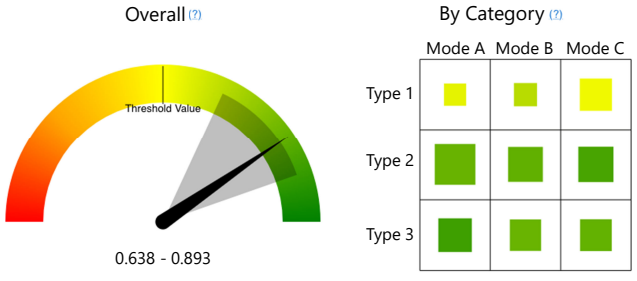

Fig. 8. Partial screenshot of Dynamo GUI for AN/TPQ-53 proxy data.

Acknowledgment

The authors thank Jeremy Werner, Sandra Hobson, Nate Crookston, Larry Stone, Tom Corwin, and Sean Daugherty for their support of this work and their contributions to it.

References

[1] J. Ferry, L. Stone, T. Corwin, A. Ahmed, S. Daugherty, J. Werner, and S. Hobson, "Use of Bayesian methods to optimize decisions," *Naval Engineers Journal*, vol. 136, no. 1, pp. 79–84, 2024.

[2] R. A. Fisher, *The Design of Experiments*. Oliver and Boyd, 1935.

[3] D. C. Montgomery, *Design and Analysis of Experiments, 8th ed.* Hoboken, NJ: John Wiley and Sons, 2012.

[4] E. T. Jaynes, *Probability Theory: The Logic of Science (Vol 1)*. Cambridge University Press, 2003.

[5] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis, 3rd ed.* CRC Press, 2013.

[6] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis*. New York: Springer-Verlag, 1985.

[7] L. D. Stone, R. L. Streit, and S. L. Anderson, *Introduction to Bayesian Tracking and Particle Filters*. Cham, Switzerland: Springer Nature, 2023, vol. 126.

[8] K. Chaloner and I. Verdinelli, "Bayesian experimental design: A review," *Statistical Science*, vol. 10, pp. 273–304, 1995.

[9] T. Rainforth, A. Foster, D. R. Ivanova, and F. B. Smith, "Modern Bayesian experimental design," *arXiv:2302.14545*, 2022.

[10] S. Greenhill, S. Rana, S. Gupta, P. Vellanki, and S. Venkatesh, "Bayesian optimization for adaptive experimental design: A review," *IEEE Access*, vol. 8, pp. 13 937–13 948, 2020.

[11] A. Wald, *Statistical Decision Functions*. New York: John Wiley and Sons, 1950.

[12] L. J. Savage, "The theory of statistical decision," *Journal of the American Statistical association*, vol. 46, no. 253, pp. 55–67, 1951.

[13] ——, *The Foundations of Statistics*. New York: John Wiley and Sons, 1954.

[14] D. V. Lindley, "On a measure of the information provided by an experiment," *Ann. Math. Statistics*, vol. 27, pp. 986–1005, 1956.

[15] ——, *Making Decisions*. London: John Wiley and Sons, 1971.

[16] H. Raiffa and R. Schlaifer, *Applied Statistical Decision Theory*. Division of Research, Harvard Business School, 1961.

[17] R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

[18] D. Bertsekas, *Dynamic programming and optimal control: Volume I, 4th ed.* Athena scientific, 2012.

[19] H. Chernoff, *Sequential Analysis and Optimal Design*. Philadelphia: SIAM, 1972.

[20] A. Wald, "Sequential tests of statistical hypotheses," *The Annals of Mathematical Statistics*, vol. 16, no. 2, pp. 117–186, 1945.

[21] M. Lewis, *Moneyball: The Art of Winning an Unfair Game*. New York: W. W. Norton and Company, 2003.

[22] J. Ferry, "Experimental design for operational utility," *The ITEA Journal of Test and Evaluation*, vol. 44, no. 3, 2023.

[23] D. Williams, *Probability with Martingales*. Cambridge University Press, 1991.