The Principled Prediction-Problem Ontology: when black box algorithms are (not) appropriate

Jordan Rodu
Department of Statistics, University of Virginia

and

Michael Baiocchi
Department of Epidemiology and Population Health, Stanford University

January 22, 2020

Abstract

Black-box algorithms have had astonishing success in some settings. But their unpredictable brittleness has provoked serious concern and increased scrutiny. For any given black-box algorithm understanding where it might fail is extraordinarily challenging. In contrast, understanding which settings are not appropriate for black-box deployment requires no more than understanding simply how they are developed. We introduce a framework that isolates four problem-features – measurement, adaptability, resilience, and agnosis – which need to be carefully considered before selecting an algorithm. This paper lays out a principled framework, justified through careful decomposition of the system components used to develop black-box algorithms, for people to understand and discuss where black-box algorithms are appropriate and, more frequently, where they are not appropriate.

Keywords: machine learning, black box, algorithms, reasoning
1 Introduction

It is exciting to witness the development of flexible, fast, and wildly useful predictive algorithms. Algorithms are successfully driving cars, identifying breast cancers, enabling globe-spanning businesses, and organizing unprecedented amounts of information. The excitement about these algorithms is warranted; these achievements are unparalleled in history. A very natural question for members of the business, government, and academic communities is: “how can we use them?” This simple question is more complicated than it first appears because modern predictive algorithms have several very distinctive features, including this one: some of the most successful algorithms are so complex that no person can describe the mathematical features of the algorithm that gives rise to the high performance (a.k.a. “black box algorithms”). There is an important tension right now because these extraordinary black-box algorithms exist – with so much potential to do good – despite deep uncertainty about when and how to use them. And this discomfort is warranted: for all of their achievements, black-box algorithms have shown to be unpredictably brittle in the real world. This is a consequence of how they are developed.

These algorithms have come into existence through a confluence of innovations. Some of these innovations are practical (e.g., the price of computing has continued to drop), some are market-based (e.g., online platforms have proved to be wildly profitable business models and so corporations have funded much of this research and development), some are due to political decisions (e.g., emphasis on STEM fields has created a big pipeline of data scientists), but a major – yet also under-appreciated – shift has come from a new framework for assessing algorithms, a framework that does not require slow-moving mathematical proofs. While understanding black-box algorithms is not possible, by carefully understanding how they are being developed and assessed we can understand what situations are more – and
less—compatible or safe for the use of black-box algorithms. To provide guidance on using prediction algorithms, this paper offers a new framework for stakeholders (e.g., business people, government officials, non-statistically minded academics) to discuss and critique the use of these algorithms.

Below we give a short introduction to the intellectual-engine that has driven much of the recent innovation, and the one that has consequences on how algorithms are deployed.

The Common Task Framework (CTF) [Liberman, 2010, Donoho, 2017, Breiman, 2001] provides a fast, low-barriers-to-entry means for researchers to settle debates about the relative utility of competing algorithms. This is in contrast to the traditional use of mathematical descriptions of the behavior of an algorithm, or simulations of the algorithm’s ability to recover parameters of a data generating function. Many readers are likely familiar with the CTF even if the name is unfamiliar; the NetFlix Prize [Bennett et al., 2007] and Kaggle competitions are excellent examples of this framework. The key features of the CTF are: (a) curated data that have been placed in a repository; (b) static data (all analysts have access to the same data); (c) a well defined task (e.g., predict y given a vector of inputs x for previously unobserved units of observation); (d) consensus on the evaluation metric (e.g., the mean squared error of the predictions from the algorithm on a set of observations); and (e) an evaluation data set with observations which have not been accessible to the analysts. Today, in practice, some of the features of the CTF are relaxed. In particular, outside of the major competitions, feature “e” is often self-policed - i.e., the analyst has direct access to the evaluation data set.

The CTF provides an efficient environment for development that allows innovation in prediction algorithms to move quickly. The data exist already. All analysts have access to these common data so many people can work on the problem at the same time. Fast computation takes the place of proving theorems, and performance is quickly assessed
using held-out data. The consequences of a poorly performing prediction algorithm in the CTF is minimal – e.g., after a failure the analyst tweaks the algorithm and tries again. Fundamentally, the CTF takes complex real-world problems and sand-boxes them.

In the CTF, because there is a specific performance metric, there is little ambiguity in the relative ordering of the algorithms conditional on a particular dataset. The underlying logic of using leader boards is new and extremely productive. Leader boards rely on a form of reasoning we call “outcome-reasoning” (discussed in detail in 4.1). For the reasons described in the previous two paragraphs, outcome reasoning – if appropriate – is preferred. However, many people are currently deploying black-box algorithms, which inherently rely on outcome-reasoning, in problem settings when outcome-reasoning is unavailable. In these problem settings, “model-reasoning” should be used. Many current debates in the literatures about the suitability of black box models hinge on (mis)understandings about what kind of reasoning is appropriate for given problems. The goal of this paper is not to resolve these debates but to provide a useful framework for understanding the type of problem for which a solution is sought.

In this paper, we make three fundamental shifts in focus from how current debates about black box algorithms usually proceed. We make these shifts explicit here, to prevent reading this paper as contradicting existing decision-making frameworks that analysts are already using to assess black box algorithms. Instead of being in contradiction, the three shifts listed below indicate that our considerations simply occur earlier in the data-driven decision-making pipeline than do existing frameworks. The three shifts are 1) Our ontology starts with stakeholders defining a prediction-problem. This means we think about how to elicit ideas and feedback from stakeholders. We work through several examples in Section 3 but we call your attention to the example on recidivism in subsection 3.4. An immediate implication of our ontology’s focus on stakeholders is that changes in stakeholder member-
ship is likely to change how we think about the prediction-problem. 2) We focus on crafting the problem and its consequences. This means we are engaging the problem quite early in the process - helping to understand, shape, and quantify the key issues. We do not start after the “prediction task” already exists; it is not given to us. Like in experimental design, the stakeholders must first think about what they want to accomplish by using data, and then we help to turn that into a specification of the problem. 3) Our ontology focuses on features of the problem, rather than features of the algorithm. Problem features are a consequence of both real world constraints, and the interplay between those constraints and stakeholders’ goals (see section 3 for detailed examples).

We designed this paper to be accessible to different readers. Policy-minded readers, scientists, and non-technical academics will find the information most useful to them from sections 2 and 3. Analysts and technical academics who are asked by colleagues to select appropriate algorithms will hopefully find sections 2, 4, and 5 provide useful context and language to guide these conversations. This modularity leads to some slight repetition between sections, but even those in the algorithm research and development communities will benefit from the reinforcement of key concepts.

2 Introduction to the Principled Prediction-Problem Ontology

In any interesting prediction problem, errors will occur. Viewed one way, the Prediction-Problem Ontology is a framework for achieving buy-in from stakeholders before these errors start to accumulate. Imagine for a moment that a black box algorithm is shown to outperform all other existing algorithms in the training and test data. Based on this information alone, the algorithm is deployed. After deployment a terrible event occurs due to errors in
the predictions. Who is accountable for the consequences of these errors? The reasoning
that went into the algorithm’s justification turned on knowledge owned by the analyst:
Were the training and test data sets adequate? Were the algorithms selected for consid-
eration adequate? Was the performance metric informative? Did the analyst adequately
protect against over-fitting? These are technical questions. How could stakeholders be held
accountable if the algorithm cannot be interrogated by other means? So who is accountable
in this situation? The decision-making turned on trust in the analyst’s judgment.

We can avoid this dynamic by actively include non-technical stakeholders in the decision-
making process. Making use of key features of the Common Task Framework, and insights
from statistical modeling, we propose the Principled Prediction-Problem Ontology. As the
ontology is derived from the CTF, and given other obvious considerations, we abbreviate
using C-3PO.

In C-3PO we classify problems using four features (“problem-features”):

1. [measurement] ability to measure a function of individual predictions and actual
   outcomes on future data,

2. [adaptability] ability to adapt the algorithm on a useful timescale,

3. [resilience] tolerance for accumulated error in predictions, and

4. [agnosis] tolerance for potential incompatibility with stakeholder beliefs.

We refer to the four problem features collectively with the mnemonic “MARA.” Stake-
holders classify their problem as either “satisfying” or “not satisfying” each problem-feature
individually. (In some settings it may be more appropriate to relax the binary classifica-
tion.) If a problem satisfies the MARA problem-features then the problem is suitable
for outcome-reasoning – that is, the powerful form of reasoning that is the foundation
for the CTF. If the problem fails to satisfy even one of the features then the problem requires a more complex form of reasoning to justify the algorithm’s deployment – i.e., model-reasoning (discussed in detail in 4.1).

Note that the curation process that goes into creating a problem for use in the traditional CTF – i.e., abstracting data sets from their applied example, providing labeled outcomes of interest, and standardizing the task and performance metric – ensures satisfying the MARA problem features. Algorithms that are developed in the CTF can be successfully applied to problems in the real world that similarly satisfy the MARA problem-features. However, these algorithms may be unsuitable for deployment in problems that do not satisfy the problem-features. C-3PO provides language to clarify a problem’s features and facilitate debate among stakeholders about the suitability of an algorithm.

The scenario sketched out at the beginning of this section is extreme – that is, a “terrible event” comes about from missed predictions. This motivation focuses attention on the resilience feature. This kind of concern has provided the main entry point for many of the critiques of using black-box algorithms in the real-world. While failures to properly consider resiliency before deploying points to very real concerns, tolerance for prediction error is only one class of concerns. There are concerns that do not hinge at all on the consequences of the errors (e.g., agnosis). Our framework both describes and provides language for engaging these concerns.

Two dynamics contribute to this framework. First, C-3PO extends the CTF to “live” problems, providing a principled foundation for assessing the generalizability and transportability of an algorithm into the real-world. Second, and most fundamentally, C-3PO starts with the stakeholders - that is, the people who are accountable to the performance of the algorithm. Starting with stakeholders has two major implications: (a) the focus of the evaluation is based on understanding the problem itself rather than the algorithm, and
(b) if different stakeholders are brought in and out of the group assessing the problem then the group can reach very different conclusions about the appropriateness of a black box algorithm (see the recidivism example below for how this works).

This framework implies a workflow (see Fig 1) in which stakeholders first engage a prediction problem through C-3PO. This occurs prior to any technical considerations (e.g., using cross validation to assess the fit of the algorithm). Only after classification under C-3PO, analysts can identify suitable algorithms that adhere to the proper form of algorithmic reasoning, and can assess the potential algorithms for their technical merits. Finally the analyst can deploy the algorithm with proper re-assessment, depending again on the class of reasoning chosen through C-3PO.

2.1 Stakeholders

One of the fundamental tenets of C-3PO is that it focuses energy in decision making on the problem at hand. This inevitably forces the question of who is defining the problem, or who is affected by the problem. Loosely, these are the “stakeholders.” Until this point, we have avoided specifying who the stakeholders are, as this largely depends on the context of
the scenario under which the problem is formulated. But there is a tension between being too specific on the one hand and pigeonholing this manuscript in one domain or another on the other hand. In this subsection we spend a few paragraphs working through problems and focus on who stakeholders might be. The notion of the stakeholder is quite flexible and requires explicit consideration.

First, note that the definition of a stakeholder is contingent on several decisions. For instance, a hospital considering the adoption of software to automate or augment a task might very well consider the stakeholders to be the upper management who oversee the overall financial health of their institution. However, they might also consider the doctors or nurses as stakeholders, who might bring additional constraints to considerations of satisfaction of MARA. In some cases, the hospital might wish to consider a hypothetical patient as a stakeholder, who would in many scenarios impact how we think about the adaptability problem-feature of C-3PO. C-3PO does not prescribe who is a stakeholder. Rather, once stakeholders have been identified, C-3PO aides in the discussion of the merits of various algorithms.

Now consider how a business might define stakeholders in several ways. In some scenarios, the stakeholders might be the C-suite, who are responsible for the big decisions a company has to make. Alternatively, a business might engage employees at any level, or even customers in understanding their problem’s satisfying of MARA constraints. And even the notion of customer may have different interpretations here – current customers as well as future customers, which may diverge in their interests. It’s likely that in many settings, in order to anticipate more long-term consequences of their decisions, the company will need to include shareholders as stakeholders.

While the term “stakeholder” might conjure a business application or community mobilization, we intend for it to refer to a broad range of disciplines, including academia. A
scientist, for instance, should define stakeholders when engaging C-3PO. In some cases, the stakeholder will be the scientist. In other cases, the scientist might wish to include her peers as hypothetical stakeholders, perhaps representing the pursuit of knowledge in a broad sense. In some cases it might make sense for the scientist to consider journal editors as stakeholders. With an eye towards getting published, the scientist might put more weight on the agnosis problem-feature when considering editors as stakeholders, for instance, than she otherwise might.

C-3PO defines a framework for engaging decisions on deploying algorithms. There are two levels to this debate. The first is on the scope of the stakeholders. The second is on the satisfaction of MARA, given the stakeholders. While we anticipate a lot of time and emphasis to be placed on whether or not a problem-feature is satisfied, it is critically important to debate who should be included as stakeholders as inclusion/exclusion will greatly impact the assessment of the MARA problem-features.

One of the most important examples of stakeholder involvement arises in government. Governing committees often seek involvement and input from a cross-section of the population, with particular emphasis on those most impacted by decisions. In large part, C-3PO was developed with this kind of dynamic in mind: bridging the communication-gap between non-technical stakeholders and analysts. While stakeholders may benefit from the use of these algorithms, they should not be required to understand the technical issues. With the appropriate framework, stakeholders can offer useful critiques and assessments of the prediction-problem which can be incorporated into the design of the study.
3 Examples

In this section we discuss several common prediction problems. Each example was chosen to highlight different aspects of C-3PO. Each of the MARA problem-features is considered, as well as how stakeholder selection impacts these considerations.

3.1 Recommendation systems

The goal of a recommendation system is to introduce users to products of interest. Training data for a recommendation system often is represented as a sparse matrix with individuals on the rows and products on the columns. If the $i^{th}$ user has engaged with the $j^{th}$ product and rated it, the $ij$ entry of the matrix will contain the user-product rating. Other entries will be blank. For held-out data, some of the entries are obscured, and the task is to predict the rating for those user-product pairs. In deployment, algorithmic performance can be measured as a function of the individual outcome of each recommendation, which might take the form of a) acceptance of recommendation with a subsequent high rating, b) acceptance with a low rating, c) acknowledgement of the recommendation without taking it, or d) no acknowledgement of the recommendation. Technically, only some of the predictions are verified in the wild. When the system predicts that the user will assign a low rating to a product, that product will not be recommended, and that rating might not be verified. But the ultimate goal of the recommendation system is to provide good recommendations, ultimately to, say, make money. Often there is an extremely large pool of products, many of which could be recommended to the user with great success. The loss function, then, is asymmetric. If a product is recommended to a user, it is desirable that the product will obtain a high rating. On the other hand, if a product that would otherwise have been enjoyed by a user is not recommended, because of the large pool of potential products,
missing this product is not such a big deal. In general, recommendation systems satisfy the MARA problem-features, though of course one could construct specific examples in which some of the problem-features might fail to be satisfied.

### 3.2 Financial trading

Consider a high-frequency trading group with large reserves and a goal of profit maximizing. In order to construct a risk-diversified portfolio one sub-goal in high frequency trading is to predict the instantaneous covariance between stocks. While one cannot directly assess the accuracy of the covariance prediction, one could use an external measure of performance as a direct consequence of the prediction to monitor the success of the algorithm. For example, the group can monitor if the ongoing use of the algorithm increases the value of their holdings. In this setting, all four problem-features are satisfied.

In contrast, a manager of family wealth may require an algorithm that can be evaluated by the family so that they can assess whether or not the algorithm’s anticipated behavior matches their beliefs about the market or to verify that the trading algorithm comports with their ethical concerns. In this case, agnosis would not be satisfied.

### 3.3 Prediction in lieu of measurement

An interesting application of prediction algorithms is to use them as cheap measurements in lieu of obtaining expensive, gold-standard labels. Consider an automated system for triaging mild health symptoms. Instead of using a telephonic nurse-based system, automated prediction algorithms could be used to offer some level of diagnosis and either recommend a patient seeks further help or not. From the perspective of a health care administrator, triaging mild health symptoms may satisfy all four problem-features. However, if the pa-
tient is included as a stakeholder, then the adaptability problem-feature is violated since this is a one-shot prediction for this patient. This is a general principle: problem-feature satisfaction depends on who is considered as a stakeholder.

3.4 Recidivism

In trying to predict recidivism, if the algorithm is used such that all defendants with a score above a certain threshold are incarcerated, we are unable to observe the correctness of our predictions for people who are incarcerated. This arises from a missingness in the outcome, and will happen in general when the prediction algorithm causes changes in the outcome. The recidivism problem violates the measurement, resilience, and agnosis problem-features. In the case of the measurement problem-feature, we are not able to observe the outcome for a portion of the defendants. In the resilience problem-feature, the debate hinges on the stakeholders’ concerns about depriving rights through unnecessary incarceration, balanced against possible future criminal acts. Failure to satisfy agnosis stems from two concerns. First, it is necessary to explain to the defendant why the decision to incarcerate was made. Second, even if the algorithm were a flawless predictor, if it did so through morally repugnant means, this would raise deontological ethical concerns that would need to be debated by stakeholders.

3.5 Optimizing causal predictions

Say we run a website that can place only one of three ads – corresponding to one of three items for purchase – for each customer who arrives to the website. We are unsure of which ad to place for a given customer. Let us consider if a black box algorithm is suitable for use in learning an optimal assignment of ads.
In this setting, a useful algorithm estimates how much the probability of buying a product changes given that the website shows a given ad to a given customer. At first pass it may seem best to assign customers to the ad that will increase the probabilities of a sale the most – but that is true only if the analyst knows how the probabilities of purchasing change given an ad, perhaps up to some tolerance.

When trying to optimize a causal prediction algorithm, we have to assign some observational units such that for any type of customer there is some positive probability of receiving a given ad. This requirement is called the positivity assumption. Informally, the positivity assumption requires that for any person in the study there is a positive probability to receive any of the treatments – and in practice this is useful for estimating causal estimates because (with enough data) the positivity assumption ensures there are people who are quite similar who can be used as contrasts. If positivity does not hold (e.g., we assign all customers with a high estimated probability of purchasing to receive the ad) then we can no longer estimate the causal effect of placing the ad. This is the intuition behind explore-exploit algorithms, particularly contextual bandits. In part, the recidivism example above failed the measurement feature because of its lack of positivity.

For this example the important information is the differences between how each of the ads changes a given customer’s probability of purchasing. But this quantity is never observable because whenever we assign a customer to a particular ad we obscure the ability to observe what would have happened if we had shown this customer one of the other two ads (this is known as the “Fundamental Problem of Causality”). It may feel like in this example we fail the measurement feature because we can’t observe the desired quantity (i.e., the difference in in the change in probabilities). Yet this is incorrect because the measurement feature allows functions of the predictions to satisfy the feature. We can consider a function that uses the product of the item-specific outcome and the (binary)
treatment assignment. This is a common trick in causal inference (e.g., sum the outcomes of the treated and subtract the sum of the outcomes of the control), and gives rise to methods based on propensity scores, and randomization based tests. With this function, we can obtain all of the required observable quantities to verify the function. In fact if the assignment mechanism is designed by the researchers and contains some randomness then this problem takes on the structure of a randomized controlled trial.

In most randomized controlled trials (RCTs), the goal is to directly assess the compatibility of beliefs with reality, thus traditional RCTs fail agnosis. When human subjects are the subjects in RCTs, issues related to resilience are considered by the institutional review board. But when the MARA problem-features are satisfied, this gives rise to a peculiar problem type: an atheoretic randomized controlled trial with treatment assignment determined by a black box algorithm.

4 Reasoning about algorithms

In this section we describe the types of reasoning that can be used for assessing an algorithm.

4.1 Model- and Outcome-reasoning

C-3PO concentrates on the degree to which beliefs, or the current state of knowledge in the given domain, are used to constrain a model. While content knowledge is rarely ignored entirely, its utilization in assessment can vary from heuristically informing the choice of method to actively validating the learned parameters. We now introduce two forms of assessment that reside on either end of this spectrum: model-reasoning and outcome-reasoning.

Model-reasoning requires checking that the model conforms to current beliefs. For
example consider a linear regression; we can use model-reasoning by verifying the direction of individual coefficients matches what is expected from our domain knowledge. We can think of these checks as a mapping from the model, or “parameters” of the model, to the space of current beliefs. In model-reasoning, it is therefore possible to hypothesize how a particular instantiation of a model, \( \hat{f} \), will perform on future data without reference to the data used to fit the algorithm. This provides solid ground on which experts in a field can debate and discuss the fitted algorithm and its suitability for future predictions in a concise manner, with discussions stemming from beliefs, and not potential difficult-to-find shortcomings of the data set or algorithm.

In contrast, outcome-reasoning relies very little on beliefs, which primarily enter into consideration through choice of the performance metric. Outcome-reasoning is the reasoning used in the CTF. In C-3PO, outcome-reasoning is extended to the ongoing, out-of-sample prediction setting. If the four problem-features are satisfied then the analyst and stakeholders can monitor the performance metric during the deployment phase in order to assess whether or not their algorithm is working.

The two types of reasoning lead to two different kinds of thinking when comparing algorithms. Model-reasoning tends to involve discussions of parameters and the algorithm’s ability to faithfully recover the parameters. That is, model-reasoning forces the analyst to think carefully about how changes in the covariates should be linked to variation in the outcome (e.g., should we predict that a taller person weighs more than a shorter person?). But when wildly different algorithms are compared – e.g., say an ARIMA\((p,q)\) is compared to a decision tree – it is quite challenging to translate between divergent conceptualizations of how the input space is linked to the outcome space. Consequently, given the challenge of translating between algorithms using model-reasoning, comparisons tend to be pairwise and slow. In stark contrast, outcome-reasoning assiduously avoids any debates in the input
space. Instead, outcome-reasoning exclusively operates in the space of the outcome—where all candidate algorithms must operate. An analogy to capture this dynamic: consider two economies. Model-reasoning is a bit like a barter-based economy; each transaction requires careful consideration of idiosyncratic features and how much the parties need each of the products. An economy that uses currency to store value allows a lower friction form of transactions; each product’s value is translated into the currency and then comparisons can be made rapidly between different products. Now imagine these two economies and their ability to develop, innovate, and scale.

Many, though certainly not all, concerns about black box algorithms can be framed as issues of extrapolation. While the CTF offers a foundation for comparing algorithms’ performance on currently available data, the CTF alone does not offer a principled foundation for reasoning about the future, out-of-sample performance of an algorithm. Without access to model-reasoning, the mechanisms for reasoning through performance on future data, and not just held-out data, are limited. Such reasoning would require careful consideration of the interaction between properties of the data set and the properties of the algorithm. In a setting that uses a black box algorithm that require massive training data, understanding the data itself can be impossible. Even under a smaller data regime, the task of unpacking the data/algorithm interaction can be difficult, if not impossible. Indeed, as discussed in the introduction, one common fix when a black box fails is to add data to the training data set in hopes that a new fit on the new data might remedy the failure. The underlying cause for the failure with respect to the current \( \hat{f} \) often remains unknown.

An important distinction between the two modes of reasoning: model-reasoning allows for detailed debate to happen before the deployment of the algorithm, whereas outcome-reasoning affords assessment purely post-deployment.
4.2 The Ontology

We now discuss the four problem-features in more detail. Collectively, we refer to the four problem-features as MARA.

4.2.1 Problem-feature 1: measurement

The first problem-feature is the ability to measure a function of the predictions and actual outcomes on future data. Let $y^*$ be the value of a future outcome associated with predictors $x^*$, and $\hat{f}(x)$ denote the estimated prediction function. For some agreed-upon notion of close, this problem-feature describes the ability to track whether $\hat{f}(x^*)$ is close to $y^*$, measured as $g(y^*, \hat{f}(x^*))$, within some reasonable tolerance.

This is the most foundational problem-feature for the C-3PO framework. If this problem-feature isn’t satisfied the analyst will not be able to verify if the algorithm is performing well. The use of a black box model for a problem that doesn’t satisfy measurement requires faith. If this problem-feature is satisfied then the algorithm’s performance can be monitored after deployment by monitoring the error function $g()$ (notably, both Google [Google, 2019] and Uber [Hermann et al., 2018] include monitoring predictions after algorithm deployment as a critical component of their machine learning workflows). Without this feedback mechanism, assessment must happen before deployment. In this case, our framework requires that the analyst pursue model-reasoning.

4.2.2 Problem-feature 2: adaptability

Problem-feature 2 is the ability to adapt the algorithm on a useful timescale. In some settings, upon discovering errors in prediction, an algorithm can be updated relatively quickly and will be presented with sufficient opportunity to update. In other settings,
the underlying dynamics of the population of interest change at a rate such that those changes dominate algorithm adaptations from observed error. The latter situation renders predictions as one-shot extrapolations, at which point the observation of the function $g()$ is useless.

For instance, predicting the outcome of the United States presidential election depends heavily on measuring the ebb and flow of priorities of the voting population. With an algorithm assessed under outcome-reasoning, the lessons learned from prediction errors in one election may not be informative for the next election because the underlying priorities of the population may have shifted. Another common violation of adaptability is when the deployment of the algorithm itself changes the way the outcomes are generated; this phenomenon has been described many times in policy settings - the Lucas Critique, Goodhart’s law, and Campbell’s law being famous formulations.

4.2.3 Problem-feature 3: resilience

Problem-feature 3 describes the stakeholders’ tolerance for accumulated error in predictions. As errors accumulate, someone or some group will be held accountable. Some stakeholders will see errors in prediction as so intolerable as to bar any unjustified use of an algorithm, for example when an error in prediction may lead to a death or a false incarceration. On the other end, settings like recommendation algorithms may be viewed as having minimal consequences to errors in prediction. Most scenarios will be somewhere in between, where the stakeholders are willing to trade off some unaccounted error in predictions against the accumulation of value gained from better predictions. If the group deploying the algorithm has large reserves relative to the accumulation of costs due to the accumulation of errors then the problem at hand satisfies resilience.
4.2.4 Problem-feature 4: agnosis

Problem-feature 4 describes tolerance for incompatibility with stakeholder beliefs. Stakeholders will hold certain beliefs about the process being predicted. These beliefs may take the form of prior knowledge or scientific evidence (e.g., experience with how gravity works in this setting). Other beliefs may arise from moral or ethical concerns (e.g., racial information should not be used to assign credit scores). In some cases, if an algorithm reaches its predictions in a way that violates their beliefs then this dynamic can make stakeholders uncomfortable with deploying the algorithm.

The agnosis problem-feature requires both eliciting and clarifying the stakeholders’ beliefs about the problem. Understanding agnosis also requires the analyst and stakeholders to gauge comfort with the algorithm violating their beliefs.

4.3 Problem-feature discussion

To make concrete the selection of model-reasoning or outcome-reasoning, we provide a decision tree that links the satisfaction of the problem-features with the type of reasoning. Note that a decision to use outcome-reasoning requires traversing the leftmost branch of the tree.

In Section 3 we took great care to isolate examples so that each problem-feature appeared as clear and distinct as possible. In reality, these features interact and in practice should be discussed collectively as MARA.

Model-reasoning requires deductive reasoning, meaning that we understand the mathematical structures of the model well enough so that once decoupled from the data it was fit on, stakeholders can reason about future behavior of the algorithm. Methods of assessing an algorithm that are inductive cannot be used for model-reasoning. Inductive reasoning
is necessarily contingent and depends on the data in hand (e.g. recycled predictions) or on details of hypothesized, future, out-of-sample data. Methods of assessment built off of these are attempting to approximate model-reasoning.

5 Perspective on Black Box Algorithms

In this section we try to clarify a miscommunication happening in debates about black box algorithms. To orient the discussion we look at the following question twice: “What happens if a black box algorithm fails?”

First, in a technical sense, what happens if a black box algorithm is deployed in the real world and something goes wrong? Speaking loosely, there are four fixes that are commonly used to correct an algorithm that is not well understood: 1) add a human to the loop; 2) collect more data and retrain the algorithm; 3) use a different algorithm; or 4) force the algorithm to return a pre-specified output for inputs that have been identified as producing “unacceptably bad” outputs. All four fixes might work, though they could be difficult to implement (fix 1) or impossible to validate (fixes 2-4). In the paragraph below we discuss each fix and their limitations.

The first fix is to add a human into the prediction process. For example, suppose an airplane’s autopilot algorithm malfunctions under conditions that analysts are unable to describe \textit{a priori}. While the algorithm can successfully pilot most of the time, an override can allow an alert pilot to take the controls when she detects an error, thereby inhibiting the autopilot algorithm. With the human in the loop fix, there is a fundamental loss of scalability which has been a hallmark of modern prediction. The second fix uses more training data, especially in areas of input space that the analyst believes to be problematic; however it is difficult to reason about how to sample from the space of observations. From
the perspective of a black box algorithm, it is not clear if two points are “close or “far apart” in the input space – these algorithms take advantage of highly non-linear patterns in the space of inputs. For example, suppose we have two people one who is 36 years old and another person is 36.1 years old. If we were working with an algorithm that uses smooth change in the covariate space to change its prediction then these two points might be seen to be “near” to each other and the algorithm would tend to return quite similar predictions, but there is no such guarantee of smoothness with a black box algorithm. Metaphorically, the black box algorithm “thinks” very differently about changes in the covariate space than more accessible models, so it is hard for an analyst to think about how to get “useful” or “novel” or “divergent” data to use to retrain the black box algorithm. The third fix effectively restarts the original search for an appropriate algorithm. As with the previous fix, it is impossible to say that the original failure is corrected (without actually deploying the algorithm and waiting to see if it fails again), and it is possible that new failures have been introduced. The fourth fix is technically a special case of the third fix, creating a hard patch wherein the analyst forces the algorithm to return an analyst-determined prediction in parts of the input space which are most problematic. For instance, if an image labeler pairs otherwise innocuous labels and images in a way that is objectionable, then the analyst may force the algorithm to return a non-response whenever that input is paired with that label. We reiterate that, despite these limitations, these four fixes may be entirely appropriate for specific circumstances. In other cases, these limitations may present insurmountable objections, rendering these fixes effectively useless.

Let us return to the orienting question: “What happens if a black box algorithm fails?” Miscommunication appears repeatedly in the literature because this question is used differently by different people in order to highlight different types of concerns. Inside of data analyst communities, the current decision-making frameworks and conversations about im-
proving prediction have understood this question in the technical ways outlined in the two paragraphs above. But the question should also be understood as asking: “How will responsibility be assigned?” There is real value to be gained from deploying complex algorithms in the real world but how can non-technical stakeholders be made part of the decision-making process for deployment? If we do not offer a meaningful framework for non-technical stakeholders to interrogate the suitability of an algorithm for deployment then the entire burden is on the analyst. Our community’s work on impressive statistical technologies has outpaced our work on means for including our non-technical collaborators at vital points in the development and deployment of these algorithms.

This paper is motivated by the “How will responsibility be assigned?” style of questions.

6 Related Work

Recently the CTF, while continuing to yield huge success in algorithmic development, has seen a host of criticisms. One concern about the CTF is that datasets can be overfit over time [Sculley et al., 2018, Rogers 2019, Van Calster et al., 2019, Ghosh, 2019]. Despite all attempts to protect against overfitting, idiosyncratic aspects of a particular dataset are learned when heuristic improvements yield improved predictions over the state of the art. Additionally, and a potential corollary of this criticism, given two sets of reference data, it is often unclear why an algorithm performs well on one dataset, but poorly on the other. This has led some to call for more resources to be dedicated to understanding the theoretical underpinnings of the algorithms that have achieved such huge success, hoping to avoid a potentially catastrophic failure in the future. There have been several debates on the relative merits of careful theoretical justification vs. rapid performance improvement.
We do not enter this debate here. However, the substance of the debate is important in C-3PO. As can be seen in how these algorithms respond to different datasets, their performance is a complex interaction of the data, which can often be quite large, and difficult-to-uncover aspects of the algorithms. While some in the aforementioned debates call for more theoretical understanding of these algorithms ahead of rapid innovation, we take a different tack and ask when we can reasonably deploy a black box algorithm through outcome-reasoning to make predictions in the wild. For an algorithm whose future performance is justified using a measurement of the algorithm’s success in the space of the outcome, as is done in the CTF, this framework recognizes an extension to the CTF, at very least satisfying the measurement problem-feature. In problems that do not satisfy the measurement problem-feature, the algorithm is being used to extrapolate without apriori justification, and we have no way of measuring – or perhaps even being aware of – failures. By construction, such extrapolation does not exist in the static version of the CTF.

This paper also relates to debates in ethical machine learning through both the agnosis problem feature, as well as through stakeholder inclusion. This literature is new, rapidly expanding, and impactful; we suggest interested readers consult the following as solid entry points: Corbett-Davies and Goel [2018], Lum and Isaac [2016], Kusner et al. [2017], Nabi and Shpitser [2018], Wiens et al. [2019]. We use the C-3PO framework in the examples in Section 3 to demonstrate how this framework can be used to clarify concerns of this nature – see the examples on recidivism (subsection 3.4) and prediction in lieu of measurement (subsection 3.3).

In the public literature, perhaps the most careful lines of thinking on the CTF has been undertaken by David Donoho [Donoho 2017, 2019]. (Note: it appears that much of the development of the CTF happened outside of the public-facing, academic literature.)
Particularly of interest, Donoho develops the notion of hypothetical reasoning in [Donoho 2019], exploring how analysts have developed “models” – formalizations of their beliefs into statements of probability models – to, in his words: “... genuinely allow us to go far beyond the surface appearance of data and, by so doing, augment our intelligence.” Using language developed for C-3PO we might say that satisfying the agnosis problem-feature means forgoing the advantages Donoho identifies accrue to hypothetical reasoning. That may be a reasonable choice in some settings, but it should give pause to researchers interested in generating solid scientific evidence. We strongly recommend both Donoho papers to the reader interested in understanding deeper features of the CTF as well as its historical context.

Finally, the C-3PO framework is related to work on explainability/interpretability of algorithms. Cynthia Rudin [Rudin 2018] explores the suitability of two types of models, explainable machine learning models and interpretable machine learning models, in the context of high risk and low risk predictions. In Rudin’s dichotomy, she warns against using explainable models in a high risk prediction due to our inability to make sense of the performance of the model despite the promise of explanation. Instead, for high risk scenarios, she urges the practitioners to use interpretable models that can be linked directly to domain knowledge, and encourages researchers to put effort into finding suitable interpretable models where none exist. In terms of our framework, Rudin is exploring the joint impact of the resilience and agnosis problem-features. We are convinced by Rudin’s arguments on why explainable machine learning models are not sufficient for what we identify as problems that require model-reasoning, and direct the reader to her paper for details.
7 Discussion

The C-3PO framework focuses on features of the prediction problem at hand, rather than the features of the algorithm. The problem itself is selected by the stakeholders who have concerns that include accountability. Understanding how stakeholders see the problem is the critical first step towards selecting an appropriate algorithm. This framework directs attention to the four problem-features that stakeholders should assess: measurement, adaptability, resilience, and agnosis (“MARA”). Once assessed, the appropriate method for reasoning about the algorithm can be selected.

In contrast to (but not in conflict with) C-3PO, there are other frameworks for decision-making about the suitability of an algorithm, typically technical in nature and useful for understanding the performance of different algorithms – e.g. diagnostic tools or asymptotic performance – but these are helpful largely after the method of reasoning has been selected.

While C-3PO is a statement about the problem and not the algorithm, it does imply a loose structure to the set of possible algorithms. One way to think about this implied structure is that model-reasoning methods are decoupled from the data, allowing for deductive reasoning about future performance, while outcome-reasoning relies on contingent, inductive reasoning. Many of the current approaches to describing black-box algorithms are fundamentally inductive in nature (see Rudin [2018]). While these can be quite useful, they are still a qualitatively different form of reasoning. This is a familiar distinction in the type of evidence we bring to problems, and the reader need not look hard to find examples in which deductive reasoning is a required component of our decision making. The gold-standard of inductive reasoning is randomized trials, but in the most consequential settings, the result of the most solid form of inductive reasoning does not provide sufficient justification. For example, when approving a new drug, government agencies would not
allow evidence from an atheoretical randomized trial to warrant approval. Instead agencies require a detailed scientific hypothesis about how the drug’s mechanism causes the outcome. The addition of deductive reasoning and coherence across beliefs provides a firmer, evidence-based foundation. And yet, when appropriate, the use of outcome-reasoning is to be preferred because it is an extraordinarily powerful engine for producing the highest quality predictions. Outcome-reasoning is the intellectual-engine of CTF.

If the data sets are interesting, the task is useful, and the performance metric describes an ordering that matches how the algorithm will be used then outcome-reasoning leads to an extraordinary consequence: it allows an analyst to bypass the slow, technical challenge of mathematically describing the behavior of the algorithm. Instead, outcome-reasoning allows the analyst to look at the joint distribution of predicted and observed outcomes and then rank performance of algorithms by merely creating statistical summaries. Outcome-reasoning leverages Tukeys insight that [Tukey, 1986]: “In a world in which the price of calculation continues to decrease rapidly, but the price of theorem proving continues to hold steady or increase, elementary economics indicates that we ought to spend a larger and larger fraction of our time on calculation.”

The power and popularity of the CTF has inspired extensions to prediction domains that are not traditionally investigated inside the framework. For instance in Wikle et al. [2017] the authors propose an extension to spatial prediction which, among other additions, includes an abundance of relevant data sets of differing characteristics on which the algorithm must succeed, and additional metrics, like assessment of prediction coverage. It seems reasonable that the CTF-SP will enhance rapid innovation of algorithms for certain types of problems, which is an exciting prospect for the spatial forecasting community. However, we caution that, like all algorithms that are developed in the CTF, those algorithms that are not qualified to be reasoned about using model-reasoning and should only
be used in a situation that permits outcome-reasoning.

The C-3PO framework provides a language to help stakeholders and analysts communicate the key features of a problem and then guide the selection of an appropriate algorithm. This language can also be used by algorithm developers to help identify areas for innovation. For instance, in section 3 we discuss “prediction in lieu of measurement” and we are unaware of effective algorithms that could be used to provide model-reasoning. This provides analysts and stakeholders a way to identify critical gaps in the existing set of approaches.

The unpredictable brittleness of black-box algorithms has provoked concern and increased scrutiny. But black-box algorithms have had extraordinary success in some settings. For any given black-box algorithm understanding why it might fail or when it might fail is extraordinarily challenging. In contrast, understanding which settings are appropriate for black-box deployment only requires understanding how they are developed – that is, using the Common Task Framework (CTF). The C-3PO framework extends the CTF into real-world settings, by isolating four problem-features – measurement, adaptability, resilience, and agnosis (“MARA”) – that mark a problem as being more or less suitable for black-box algorithms. We hope C-3PO will help the two cultures of statistical modeling – and our stakeholders – communicate and reason about algorithms.

8 Correspondence with Mark Liberman: historical perspective

Mark Liberman graciously agreed to read a draft of our paper. As part of his feedback, he provided a historical perspective that is often downplayed in current discussions of the CTF, but that can – as most accounts of history do – provide guidance on how we might
work inside the CTF to tackle new technical challenges [Liberman, 2020].

Liberman writes, “The [CTF] was developed as a way to manage and guide sponsored research on very hard problems that were far away from practically useful solutions – two or three decades away, as it turned out. It was NOT originally meant for the development and evaluation of real solutions to real problems, though obviously it can be (and has been) generalized in that direction.” This was intentional, and had implications on the choice of task and metric.

He says that “it was seen as a mistake to choose tasks that directly represent the real goal of the work.” Instead tasks were chosen that balanced several requirements, including their fitness for the CTF (i.e. cost-effective data set creation); their isolation of specific, current challenges that were not trivial but also not insurmountable; and their appeal to funders. “New tasks (or new versions of old tasks) should be introduced every year or so ... in order to check generalization and approach the real goal more closely.”

The same careful thinking applied to metrics. Liberman writes, “Again, it was seen as a mistake to try to measure what you actually care about.” Instead, metrics should be conceptually simple and easy to automate, and should serve to move research in the general direction of the ultimate goals– to this point, he stresses, “it was always explicit that these metrics were at best somewhat correlated with the (anyhow varied) research goals”. While tasks were to be frequently updated or changed, “metrics should not be changed very often, though new metrics need to be added from time to time as appropriate.” As examples of useful metrics in human language technology (HLT), Liberman points to “word error rate” and the BLEU metric [Papineni et al., 2002], that are flawed as metrics for real-world applications, but have served HLT research well for several decades. The key is to understand when each metric is useful in promoting progress towards the ultimate goals of research, and when they are not. We note that this notion of a task or method being useful,
but not necessarily realistic for use in the “real world”, should not seem unfamiliar, as it conjures George Box’s well-known saying, “All models are wrong, but some are useful.”

Liberman adds, “The issues in question were discussed and debated extensively in the period 1985-2005, and to a lesser extent since then.” In particular, the choices of task and metric “need to be re-thought for shorter-term applications.” The emergence of these shorter-term applications coincided with two important shifts in research drivers: 1) the technologies had become “commercially viable, so that a short-term outlook began to make sense”, and 2) “Support for R&D in these areas shifted from the government to industry.” The second point in particular has interesting implications on how the CTF is administered. Under government funding, the CTF was intentional. Tasks and metrics were heavily curated so as to promote progress as intended by the CTF, and evaluation data sets guarded by a third party. Today, in contrast, the CTF exists in flavors. Each flavor bears resemblance to the original design of the CTF, though with some characteristics heavily modified or even entirely missing.

We mention in the introduction to this paper, for instance, that often an evaluation data set is no longer maintained by a third party (of course, there are many instances, like Kaggle competitions, where this is not the case). Much research occurs in environments that rely on self-policing. The researcher will create their own evaluation data set and hold their own bake-off between their current algorithm and other algorithms. Indeed, every slight modification to an algorithm is informally compared to a previous iteration through a process that mimics the CTF. On the other hand, in some competitions, aspects of the CTF competition culture might be absent. This happens when a company creates a proprietary data set with the intention of creating an algorithm to be released as a product. In this case, the company might develop algorithms in a manner that mimics the CTF, with a task, metric, labeled data set, and rapid empirical evaluation, but with no (or at
least minimal) competition from analysts outside of the company.

We believe that C-3PO applies to all flavors of the CTF, with special focus on the technologies close to deployment (Liberman refers to these as “shorter-term”). We echo Liberman’s urging that the community carefully think through the implications that a deployment-focused CTF has on the tasks and metrics, as well implications of the relaxation or omission of characteristics of the original, carefully planned CTF. Because the deployment-focused CTF is inextricably linked to the real-world problem of interest, we stress that C-3PO should be a cornerstone in discussions.

In our original draft that we sent to Liberman, we used C-3PO to informally explore the HLT subdomain of chatbots, which try to engage a user in a natural, informative conversation, typically in a narrowly defined setting. However, compared to the rapid innovation in many natural language processing (NLP) tasks, the development and deployment of chatbots has been relatively slow. Focusing on problem-first analysis using C-3PO, it appears that the problem-feature of measurement might be difficult to satisfy, since the space of possible responses in a conversation is vast. After all, not only are there many ways to convey the same information, there are often many plausible types of information that would be appropriate for a particular response. But in the real world there is feedback on the predictions of a chatbot. A useless chatbot might garner complaints, or conversation might simply be terminated early. A cleverly designed deployment could essentially test the utility of a chatbot on an ongoing basis, serving as a proxy function to the actual measurement of interest. Assuming the other MARA problem features are satisfied (which certainly depends on the problem of interest), then outcome-reasoning could be used.

The problem for chatbot development is that, because the space of possible responses mentioned in the previous paragraph is vast, it is essentially impossible to curate a dataset for a static-CTF. Our analysis pointed to this as an odd (though possibly not unusual)
situation in which a problem is fit for outcome-reasoning, but cannot fully benefit from the friction-less environment and rapid innovation promised by the traditional CTF. One interesting consequence of this is that it might be possible to mimic characteristics of the CTF by creative early deployment of chatbots. For instance (and we caution here that this is for illustration purposes only as there are many issues we are ignoring that need to be considered), an online marketplace, let’s call it “WHAMazon”, might be interested in using chatbots to help customers determine which product best suits their needs in order to expand on the current service they offer that requires human helpers. In order to mimic the CTF through early deployment, WHAMazon could invite (or entice with discounts) customers to interact with a chatbot for two minutes in a way that reflects the customers needs at the time. Upfront, the customer would know that this was simply for fun and not intended for informational purposes. This would minimize customer frustration and potential abandonment, while providing a possibly endless amount of training data.

Given the nuanced history of the CTF that he provided, Liberman additionally suggests that there is a lack of effective choices of tasks and metrics that will lend themselves to rapid development. Adopting a development-focused (long-term) view, rather than a deployment-focused (short-term) view might help to identify suitable choices. We firmly agree with this assessment. Further, in the case where a company chooses the early-deployment strategy discussed above, the development-focus CTF vs. deployment-focus CTF distinction should still greatly affect the choice of task and metric. For instance, in a development-focused CTF design (and very much dependent on the current challenge being faced by the development team), WHAMazon could set up a scenario in which the customer was incentivized to identify, as early as possible in a conversation, if they were conversing with a chatbot or a human. In a deployment-focused CTF (once WHAMazon feels like they are close to being able to deploy their chatbot technology), they might simply
record if a customer purchased an item after having chatted with the chatbot, or requested to speak with a human after the 2 minute window had expired.

In the case of chatbots, C-3PO suggests that the CTF is a worthwhile and powerful paradigm for algorithmic development, while a careful understanding of the history of the CTF suggests that the development-focused CTF is the appropriate paradigm, at least for now. In order to harness the power of the CTF in as many domains as possible, we feel that a thorough understanding of the history of the framework is essential. Careful thinking about the CTF and the consequences of its development-focus and deployment-focus modes, combined with the considerations of C-3PO, could bring to real-world challenges a principled placement into the most effective and appropriate format for algorithm development.

9 Acknowledgments

We have many people to thank for very helpful feedback and discussions. We will be updating this section shortly.

References

Mark Liberman. Fred jelinek. Comput. Linguist., 36(4):595–599, December 2010.

David Donoho. 50 years of data science. J. Comput. Graph. Stat., 26(4):745–766, October 2017.

Leo Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). Stat. Sci., 16(3):199–231, August 2001.
James Bennett, Stan Lanning, and Others. The Netflix prize. In Proceedings of KDD cup and workshop, volume 2007, page 35, 2007.

Google. Machine learning workflow — AI platform — google cloud. https://cloud.google.com/ml-engine/docs/ml-solutions-overview, 2019. Accessed: 2019-11-6.

Jeremy Hermann, Mike Del Balso, Kåre Kjelstrom, Emily Reinhold, Andrew Beinstein, and Ted Sumers. Scaling machine learning at Uber with Michelangelo. https://eng.uber.com/scaling-michelangelo/, November 2018. Accessed: 2019-11-6.

D Sculley, Jasper Snoek, Alex Wiltschko, and Ali Rahimi. Winner’s curse? on pace, progress, and empirical rigor. ICLR Workshop, 2018.

Anna Rogers. How the transformers broke NLP leaderboards. https://hackingsemantics.xyz/2019/leaderboards/, June 2019. Accessed: 2019-11-6.

Ben Van Calster, Laure Wynants, Dirk Timmerman, Ewout W Steyerberg, and Gary S Collins. Predictive analytics in health care: how can we know it works? J. Am. Med. Inform. Assoc., August 2019.

Pallab Ghosh. Machine learning ‘causing science crisis’. BBC, February 2019.

Ali Rahimi and Ben Recht. Reflections on random kitchen sinks. http://benjamin-recht.github.io/2017/12/05/kitchen-sinks/, 2017. Accessed: 2019-11-5.

Yann LeCun. Rebuttal to “Reflections on random kitchen sinks”. https://www.facebook.com/yann.lecun/posts/10154938130592143, 2017. Accessed: 2019-11-5.

Gregory Barber. Artificial intelligence confronts a ‘reproducibility’ crisis. Wired, September 2019.
Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*, 2018.

Kristian Lum and William Isaac. To predict and serve? *Significance*, 13(5):14–19, 2016.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In *Advances in Neural Information Processing Systems*, pages 4066–4076, 2017.

Razieh Nabi and Ilya Shpitser. Fair inference on outcomes. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Jenna Wiens, Suchi Saria, Mark Sendak, Marzyeh Ghassemi, Vincent X Liu, Finale Doshi-Velez, Kenneth Jung, Katherine Heller, David Kale, Mohammed Saeed, Pilar N Ossorio, Sonoo Thadaney-Israni, and Anna Goldenberg. Do no harm: a roadmap for responsible machine learning for health care. *Nat. Med.*, 25(9):1337–1340, September 2019.

David Donoho. Comments on michael jordan’s essay the ai revolution hasn’t happened yet. *Harvard Data Science Review*, 1(1), 6 2019. doi: 10.1162/99608f92.c698b3a7. URL https://hdsr.mitpress.mit.edu/pub/rim3pvdw.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *arxiv*, 2018.

John W Tukey. Sunset salvo. *The American Statistician*, 40(1):72–76, 1986.

Christopher K Wikle, Noel Cressie, Andrew Zammit-Mangion, and Clint Shumack. A common task framework (ctf) for objective comparison of spatial prediction methodologies. *Statistics Views*, 2017.
Mark Liberman. Personal correspondence, January 2020.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics, pages 311–318. Association for Computational Linguistics, 2002.