The Challenge of Crafting Intelligible Intelligence

Daniel S. Weld[1,2]
[1]Paul G. Allen School of Computer Science & Engineering
University of Washington
Seattle, Washington

Gagan Bansal[1]
[2]Microsoft Research
Redmond, Washington

ABSTRACT

Since Artificial Intelligence (AI) software uses techniques like deep lookahead search and stochastic optimization of huge neural networks to fit mammoth datasets, it often results in complex behavior that is difficult for people to understand. Yet organizations are deploying AI algorithms in many mission-critical settings. To trust their behavior, we must make AI intelligible, either by using inherently interpretable models or by developing new methods for explaining and controlling otherwise overwhelmingly complex decisions using local approximation, vocabulary alignment, and interactive explanation. This paper argues that intelligibility is essential, surveys recent work on building such systems, and highlights key directions for research.

KEYWORDS

HCI, artificial intelligence, machine learning, interpretability

1 INTRODUCTION

Artificial Intelligence (AI) systems have reached or exceeded human performance for many circumscribed tasks. As a result, they are increasingly deployed in mission-critical roles, such as credit scoring, predicting if a bail candidate will commit another crime, selecting the news we read on social networks, and self-driving cars. Unlike other mission-critical software, extraordinarily complex AI systems are difficult to test : AI decisions are context specific and often based on thousands or millions of factors. Typically, AI behaviors are generated by searching vast action spaces or learned by the opaque optimization of mammoth neural networks operating over prodigious amounts of training data. Almost by definition, no clear-cut method can accomplish these AI tasks.

Unfortunately, much computer-produced behavior is *alien* i.e., it can fail in unexpected ways. This lesson is most clearly seen in the performance of the latest deep neural network image analysis systems. While their accuracy at object-recognition on naturally occurring pictures is extraordinary, imperceptible changes to input images can lead to erratic predictions, as shown in Figure 1. Why are these recognition systems so brittle, making different predictions for apparently identical images? Unintelligible behavior is not limited to machine learning; many AI programs, such as automated planning algorithms, perform search-based lookahead and inference whose complexity exceeds human abilities to verify. While some search and planning algorithms are provably complete and optimal, intelligibility is still important, because the underlying primitives (*e.g.*, search operators or action descriptions) are usually approximations [29]. We can neither trust nor control system behavior that we do not understand.

Despite intelligibility's apparent value, it remains remarkably hard to specify what makes a system "intelligible" or to navigate

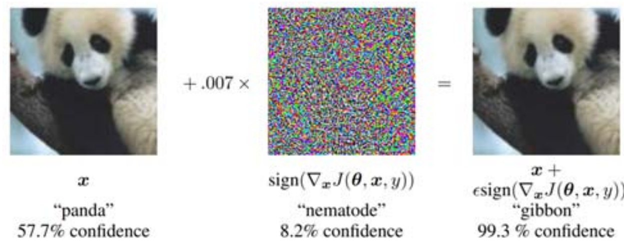

Figure 1: Figure 1 from Goodfellow *et al.* [9], demonstrating adversarial example generation applied to the GoogLeNet [39] image recognizer, trained on ImageNet. Adding an imperceptibly small vector changes GoogLeNet's classification of the image.

the tension between a concise explanation and an accurate one. We discuss desiderata for intelligible behavior later in this article. In brief, we seek AI systems where A) it is clear what factors *caused* the system's action, allowing the users to predict how changes to the situation would have led to alternative behaviors, and B) permits effective control of the AI by enabling interaction.

As shown in Figure 2, our survey focuses on two high-level approaches to building intelligible AI software: 1) ensuring that the underlying reasoning or learned model is *inherently interpretable*, *e.g.*, by learning a linear model over a small number of well-understood features, and 2) if it is necessary to use an *inscrutable* model, such as complex neural networks or deep-lookahead search, then mapping this complex system to a simpler, *explanatory model* for understanding and control [28]. Using an interpretable model provides the benefit of transparency and veracity; in theory, a user can see exactly what the model is doing. Unfortunately, interpretable methods may not perform as well as more complex ones, such as deep neural networks. Conversely, the mapping approach can apply to whichever AI technique is currently delivering the best performance, but its explanation inherently *differs* from the way the AI system actually operates. This yields a central conundrum: how can a user trust that such an explanation reflects the essence of the underlying decision and does not conceal important details? We posit that the answer is to make the explanation system *interactive* so users can drill down until they are satisfied with their understanding.

The key challenge for designing intelligible AI is communicating a complex computational process to a human. This requires interdisciplinary skills, including HCI as well as AI and machine learning expertise. Furthermore, since the nature of explanation has long been studied by philosophy and psychology, these fields should also be consulted.

This survey highlights key approaches and challenges for building intelligible intelligence. Section 2 characterizes intelligibility

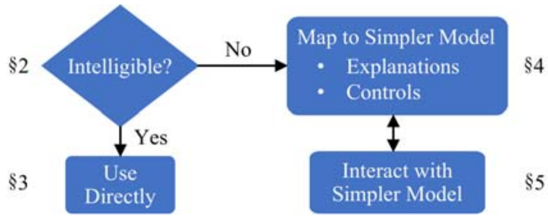

Figure 2: Approaches for crafting intelligible AI. Section numbers indicate where each aspect is discussed.

and explains why it is important even in systems with measurably high performance. Section 3 describes the benefits and limitations of GA^{2}M, a powerful class of interpretable ML models. Then, in Section 4, we characterize methods for handling inscrutable models, discussing different strategies for mapping to a simpler, intelligible model appropriate for explanation and control. Section 5 sketches a vision for building *interactive* explanation systems, where the mapping changes in response to the user's needs. Section 6 argues that intelligibility is important for search-based AI systems as well as for those based on machine learning and that similar solutions may be applied.

2 WHY INTELLIGIBILITY MATTERS

While it has been argued that explanations are much less important than sheer performance in AI systems,[1] there are many reasons why intelligibility is important. We start by discussing technical reasons, but social factors are important as well.

AI may have the Wrong Objective: In some situations, even 100% perfect performance may be insufficient, for example, if the performance metric is flawed or incomplete due to the difficulty of specifying it explicitly. Pundits have warned that an automated factory charged with maximizing paperclip production, could subgoal on killing humans, who are using resources that could otherwise be used in its task. While this example may be fanciful, it is remarkably difficult to balance multiple attributes of a utility function. For example, as Lipton observed [25], "An algorithm for making hiring decisions should simultaneously optimize for productivity, ethics and legality." However, how does one express this trade off? Other examples include balancing training error while uncovering causality in medicine and balancing accuracy and fairness in recidivism prediction [12]. For the latter, a simplified objective function such as accuracy combined with historically biased training data may cause uneven performance for different groups (*e.g.*, people of color). Intelligibility empowers users ability to determine if an AI is right for the right reasons.

AI may be Using Inadequate Features: Features are often correlated, and when one feature is included in a model, machine learning algorithms extract as much signal as possible from it, indirectly modeling other features that weren't included. This can lead to problematic models, as illustrated by Figure 4b (and described in the next Section), where the ML determined that a patient's prior history of asthma (a lung disease) was *negatively* correlated with death by pneumonia, presumably due to correlation with (unmodeled) variables, such as these patients receiving timely and aggressive therapy for lung problems. An intelligible model helps humans to spot these issues and correct them, *e.g.*, by adding additional features.

Distributional Drift: A deployed model may perform poorly *in the wild*, *i.e.*, when a difference exists between the distribution which was used during training and that encountered during deployment. Furthermore, the deployment distribution may change over time, perhaps due to feedback from the act of deployment. This is common in adversarial domains, such as spam detection, online ad pricing, and search engine optimization. Intelligibility helps users determine when models are failing to generalize.

Facilitating User Control: Many AI systems induce user preferences from their actions. For example, adaptive news feeds predict which stories are likely most interesting to a user. As robots become more common and enter the home, preference learning will become ever more common. If users understand why the AI performed an undesired action, they can better issue instructions that will lead to improved future behavior.

User Acceptance: Even if they don't seek to change system behavior, users have been shown to be happier with and more likely to accept algorithmic decisions if they are accompanied by an explanation [18]. After being told that they should have their kidney removed, it's natural for a patient to ask the doctor why — even if they don't fully understand the answer.

Improving Human Insight: While improved AI allows automation of tasks previously performed by humans, this is not their only use. In addition, scientists use machine learning to get insight from big data. Similarly, the behavior of AlphaGo [35] has revolutionized human understanding of the game. Intelligible models greatly facilitate these processes.

Legal Imperatives: The European Union's GDPR legislation decrees the right to an explanation, and other nations may follow. Furthermore, assessing legal liability is a growing area of concern; a deployed model (*e.g.*, self-driving cars) may introduce new areas of liability by causing accidents unexpected from a human operator, shown as "AI-specific error" in Figure 3. Auditing such situations to assess liability, requires understanding the model's decisions.

So far we have treated intelligibility informally. Indeed, few computing researchers have tried to formally define what makes an AI system interpretable, transparent, or intelligible [6], but one suggested criterion is *human simulatability* [25]: can a human user easily predict the model's output for a given input? Using this definition, sparse linear models are more interpretable than dense or non-linear ones.

Philosophers, such as Hempel and Salmon, have long debated the nature of explanation. Lewis [23, p 217] summarizes: "To explain an event is to provide some information about its causal history." But many causal explanations may exist. The fact that event C causes E is best understood relative to an imagined counterfactual scenario, where absent C, E would *not* have occurred; furthermore, C should be *minimal*, an intuition known to early scientists, such as William of Occam, and formalized by Halpern and Pearl [11].

Defining Intelligibility: Following this logic, we suggest that a better criterion than simulatability is the ability to answer *counterfactuals*, aka "what-if" questions. Specifically, we say that a model is intelligible to the degree that a human user can predict how a *change* to a feature, *e.g.*, a small increase to its value, will *change* the model's output and if they can reliably modify that response curve.

[1] 2v2 Debate: Caruana, Simard vs. Weinberger, LeCun. Interpretable ML Symposium, NIPS 2017

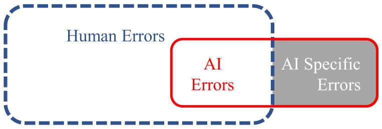

Figure 3: The dashed blue shape indicates the space of possible mistakes humans can make. The red shape denotes the AI's mistakes; its smaller size indicates a net reduction in the number of errors. The gray region denotes AI-specific mistakes a human would never make. Despite reducing the total number of errors, a deployed model may create new areas of liability (gray), necessitating explanations.

Note that if one can simulate the model, predicting its output, then one can predict the effect of a change, but not vice versa.

Linear models are especially interpretable under this definition because they allow the answering of counterfactuals. For example, consider a naive Bayes unigram model for sentiment analysis, whose objective is to predict the emotional polarity (positive or negative) of a textual passage. Even if the model were large, combining evidence from the presence of thousands of words, one could see the effect of a given word by looking at the sign and magnitude of the corresponding weight. This answers the question, "What if the word had been omitted?" Similarly, by comparing the weights associated with two words, one could predict the effect on the model of substituting one for the other.

Ranking Intelligible Models: Since one may have a choice of intelligible models, it is useful to consider what makes one preferable to another. Social science research suggests that an explanation is best considered a social process, a conversation between explainer and explainee [15, 30]. As a result, Grice's rules for cooperative communication [10] may hold for intelligible explanations. Grice's maxim of *quality* says be truthful, only relating things that are supported by evidence. The maxim of *quantity* says to give as much information as is needed, and no more. The maxim of *relation*: only say things that are relevant to the discussion. The maxim of *manner* says to avoid ambiguity, being be as clear as possible.

Miller summarizes decades of work by psychological research, noting that explanations are *contrastive*, *i.e.*, of the form "Why P rather than Q?" The event in question, P, is termed the *fact* and Q is called the *foil* [30]. Often the foil is not explicitly stated even though it is crucially important to the explanation process. For example, consider the question, "Why did you predict that the image depicts an indigo bunting?" An explanation that points to the color blue implicitly assumes that the foil is another bird, such as a chickadee. But perhaps the questioner wonders why the recognizer did not predict a pair of denim pants; in this case a more precise explanation might highlight the presence of wings and a beak. Clearly, an explanation targeted to the wrong foil will be unsatisfying, but the nature and sophistication of a foil can depend on the end user's expertise; hence, the ideal explanation will differ for different people [6]. For example, to verify that an ML system is fair, an ethicist might generate more complex foils than a data scientist. Most ML explanation systems have restricted their attention to elucidating the behavior of a binary classifier, *i.e.*, where there is only one possible foil choice. However, as we seek to explain multi-class systems, addressing this issue becomes essential.

Many systems are simply too complex to understand without approximation. Here, the key challenge is deciding which details to omit. After long study psychologists determined that several criteria can be prioritized for inclusion in an explanation: necessary causes (vs. sufficient ones); intentional actions (vs. those taken without deliberation); proximal causes (vs. distant ones); details that distinguish between fact and foil; and abnormal features [30].

According to Lombrozo, humans prefer explanations that are simpler (*i.e.*, contain fewer clauses), more general, and coherent (*i.e.*, consistent with what the human's prior beliefs) [26]. In particular, she observed the surprising result that humans preferred simple (one clause) explanations to conjunctive ones, even when the probability of the latter was higher than the former [26]. These results raise interesting questions about the purpose of explanations in an AI system. Is an explanation's primary purpose to convince a human to *accept* the computer's conclusions (perhaps by presenting a simple, plausible, but unlikely explanation) or is it to *educate* the human about the most likely true situation? Tversky, Kahneman, and other psychologists have documented many cognitive biases that lead humans to incorrect conclusions; for example, people reason incorrectly about the *probability* of conjunctions, with a concrete and vivid scenario deemed more likely than an abstract one that strictly subsumes it [16]. Should an explanation system *exploit* human limitations or seek to *protect* us from them?

Other studies raise an additional complication about how to communicate a system's uncertain predictions to human users. Koehler found that simply presenting an explanation for a proposition makes people think that it is more likely to be true [18]. Furthermore, explaining a fact in the same way as previous facts have been explained amplifies this effect [36].

3 INHERENTLY INTELLIGIBLE MODELS

Several AI systems are inherently intelligible, and we previously observed that linear models support counterfactual reasoning. Unfortunately, linear models have limited utility because they often result in poor accuracy. More expressive choices may include simple decision trees and compact decision lists. To concretely illustrate the benefits of intelligibility, we focus on *Generalized additive models* (GAMs), which are a powerful class of ML models that relate a set of features to the target using a linear combination of (potentially nonlinear) single-feature models called *shape functions* [27]. For example, if y represents the target and $\{x_1, \ldots .x_n\}$ represents the features, then a GAM model takes the form $y = \beta_0 + \sum_j f_j(x_j)$, where the f_is denote shape functions and the target y is computed by summing single-feature *terms*. Popular shape functions include non-linear functions such as splines and decision trees. With linear shape functions GAMs reduce to a linear models. GA^{2}M models extend GAM models by including terms for pairwise interactions between features:

$$y = \beta_0 + \sum_j f_j(x_j) + \underbrace{\sum_{i \neq j} f_{ij}(x_i, x_j)}_{\text{pairwise terms}} \quad (1)$$

Caruana *et al.* observed that for domains containing a moderate number of *semantic* features, GA^{2}M models achieve performance that is competitive with inscrutable models, such as random forests and neural networks, while remaining intelligible [4]. Lou *et al.* observed that among methods available for learning GA^{2}M models,

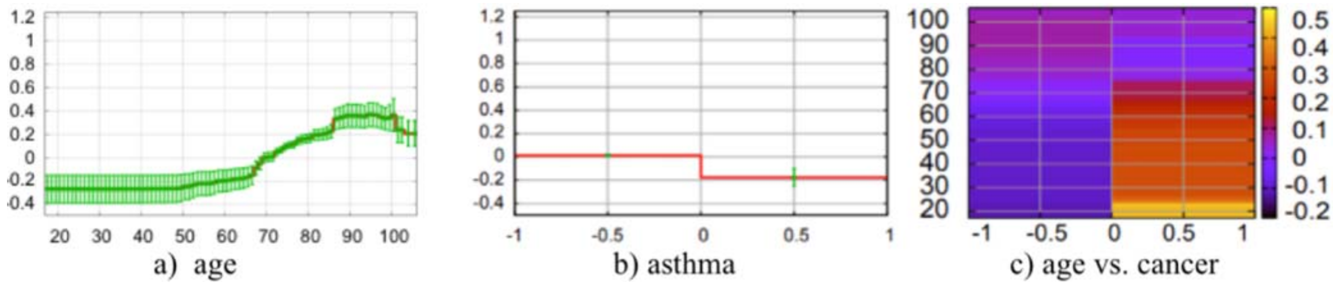

Figure 4: A part of Figure 1 from [4] showing 3 (of 56 total) components for a GA^{2}M model, which was trained to predict a patient's risk of dying from pneumonia. The two line graphs depict the contribution of individual features to risk: a) patient's age, and b) boolean variable asthma. The y-axis denotes its contribution (log odds) to predicted risk. The heat map, c, visualizes the contribution due to pairwise interactions between age and cancer rate.

the version with bagged shallow regression tree shape functions learned via gradient boosting achieves the highest accuracy [27].

Both GAM and GA^{2}M are considered interpretable because the model's learned behavior can be easily understood by examining or visualizing the contribution of terms (individual or pairs of features) to the final prediction. For example, Figure 4 depicts the contribution (log odds) of a subset of terms to total risk for a GA^{2}M model trained to predict a patient's risk of dying due to pneumonia. A positive contribution increases risk, whereas a negative contribution decreases risk. For example, Figure 4(a) shows how the patient's age affects predicted risk. While the risk is low and steady for young patients (*e.g.*, age < 20), it increases rapidly for older patients (age > 67). Interestingly, the model shows a sudden increase at age 86; perhaps a result of less aggressive care by doctors for patients "whose time has come." Even more surprising is the sudden drop for patients over 100. This might be another social effect; once a patient reaches the magic "100", he or she gets more aggressive care. One benefit of an interpretable model is its ability to highlight these issues, spurring deeper analysis.

Figure 4(b) illustrates another surprising aspect of the learned model; apparently, a history of asthma, a respiratory disease, *decreases* the patients risk of dying from pneumonia! This finding is counter-intuitive to any physician, who recognizes that asthma, in fact, should in theory increase such risk. When Caruana *et al.* checked the data, they concluded that the lower risk was likely due to correlated variables — asthma patients typically receive timely and aggressive therapy for lung issues. Therefore, although the model was highly accurate on the test set, it would likely fail, dramatically underestimating the risk to a patient with asthma who had not been previously treated for the disease.

Adjusting GA^{2}Ms with Human Feedback: A domain expert can fix such erroneous patterns learned by the model by setting the weight of the asthma term to zero. In fact, GA^{2}Ms let users provide much more comprehensive feedback to the model by using a GUI to redraw a line graph for model terms [4]. An alternative remedy might be to introduce a new feature to the model, representing whether the patient had been recently seen by a pulmonologist. After adding this feature, which is highly correlated with asthma, and retraining, the newly-learned model would likely reflect that asthma (by itself) increases the risk of dying from pneumonia.

There are two more takeaways from this anecdote. First, the *absence* of an important feature in the data representation can cause *any* AI system to learn unintuitive behavior for another, correlated feature. Second, if the learner is intelligible, then this unintuitive behavior is immediately apparent, allowing appropriate skepticism (despite high test accuracy) and easier debugging.

Recall that GA^{2}Ms are more expressive than simple GAMs because they include pairwise terms. Figure 4(c) depicts such a term for the features age and cancer. This explanation indicates that among the patients who have cancer, the younger ones are at higher risk. This may be because the younger patients who develop cancer are probably critically ill. Again, since doctors can readily inspect these terms, they know if the learner develops unexpected conclusions.

Limitations: As described, GA^{2}M models are restricted to binary classification, and so explanations are clearly contrastive — there is only one choice of foil. One could extend GA^{2}M to handle multiple classes by training *n* one-vs-rest classifiers or building a hierarchy of classifiers. However, while these approaches would yield a working multi-class classifier, we don't know if they preserve model intelligibility, nor whether a user could effectively adjust such a model by editing the shape functions.

Furthermore, recall that GA^{2}Ms decompose their prediction into effects of individual terms which can be visualized. However, if users are confused about what terms mean, they will not understand the model or be able to ask meaningful "what-if" questions. Moreover, if there are too many features, the model's complexity may be overwhelming. Lipton notes that the effort required to simulate some models (such as decision trees) may grow logarithmically with the number of parameters [25], but for GA^{2}M the number of visualizations to inspect could increase quadratically. Several methods might help users manage this complexity; for example, the terms could be ordered by importance; however, it's not clear how to estimate importance. Possible methods include using an ablation analysis to compute influence of terms on model performance or computing the maximum contribution of terms as seen in the training samples. Alternatively, a domain expert could group terms semantically to facilitate perusal.

However, when the number of features grows into the millions — which occurs when dealing with classifiers over text, audio, image and video data — existing intelligible models do not perform nearly as well as inscrutable methods, like deep neural networks. Since these models combine millions of features in complex, nonlinear ways, they are beyond human capacity to simulate.

4 UNDERSTANDING INSCRUTABLE MODELS

There are two ways that an AI model may be inscrutable. It may be provided as a blackbox API, such as Microsoft Cognitive Services, which uses machine learning to provide image-recognition capabilities but does not allow inspection of the underlying model. Alternatively, the model may be under the user's control yet extremely complex, such as a deep, neural network, where a user has access to myriad learned parameters but can not reasonably interpret them. How can one best explain such models to the user?

The Comprehensibility / Fidelity Trade-Off: A good explanation of an event is both *easy to understand* and *faithful* (accurate), conveying the true cause of the event. Unfortunately, these two criteria almost always conflict. Consider the predictions of a deep neural network with millions of nodes: a complete and accurate trace of the network's prediction would be far too complex to understand, but *any* simplification sacrifices accuracy.

Finding a satisfying explanation, therefore, requires balancing the competing goals of comprehensibility and fidelity. Lakkaraju *et al.* [22] suggest formulating an explicit optimization of this form and propose an approximation algorithm for generating global explanations in the form of compact sets of if-then rules. Ribeiro *et al.* describe a similar optimization algorithm that balances precision and coverage in its search for summary rules [34].

Indeed, all methods for rendering an inscrutable model intelligible require mapping the complex model to a simpler one [28]. Several high-level approaches to mapping have been proposed.

Local Explanations: One way to simplify the explanation of a learned model is to make it relative to a single input query. Such explanations, which are termed *local* [33] or *instance-based* [20], are akin to a doctor explaining specific reasons for a patient's diagnosis rather than communicating all of her medical knowledge. Contrast this approach with the global understanding of the model that one gets with a GA^{2}M model. Mathematically, one can see a local explanation as currying — several variables in the model are fixed to specific values, allowing simplification.

Local explanations are common practice in AI systems. For example, early rule-based expert systems included explanation systems that augmented a trace of the system's reasoning — for a particular case — with background knowledge [38]. Recommender systems, one of the first deployed uses of machine learning, also induced demand for explanations of their specific recommendations; the most satisfying answers combined justifications based on the user's previous choices, ratings of similar users, and features of the items being recommended [32].

Locally-Approximate Explanations: In many cases, however, even a local explanation can be too complex to understand without approximation. Here, the key challenge is deciding which details to omit when creating the simpler explanatory model. Human preferences, discovered by psychologists and summarized in Section 2, should guide algorithms that construct these simplifications.

Ribeiro *et al.*'s LIME system [33] is a good example of a system for generating a locally-approximate explanatory model of an arbitrary learned model, but it sidesteps part of the question of which details to omit. Instead, LIME requires the developer to provide two additional inputs: 1) a semantically meaningful set of features X' that can be computed from the original features, and 2) an interpretable learning algorithm, such as a linear classifier (or a GA^{2}M), which it uses to generate an explanation in terms of the X'.

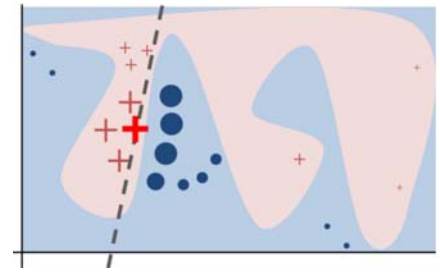

Figure 5: The intuition underlying LIME's method for constructing an approximate local explanation taken from [33]: "The black-box model's complex decision function, f, (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances, gets predictions using f, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanation that is locally (but not globally) faithful."

The insight behind LIME is shown in Figure 5. Given an instance to explain, shown as the bolded red cross, LIME randomly generates a set of similar instances and uses the blackbox classifier, f, to predict their values (shown as the red crosses and blue circles). These predictions are weighted by their similarity to the input instance (akin to *locally-weighted regression*) and used to train a new, simpler intelligible classifier, shown on the figure as the linear decision boundary, using X', the smaller set of semantic features. The user receives the intelligible classifier as an explanation. While this *explanation model* [28] is likely a poor global representation of f, it is hopefully an accurate local approximation of the boundary in the vicinity of the instance being explained.

Ribeiro *et al.* tested LIME on several domains. For example, they explained the predictions of a convolutional neural network image classifier by converting the pixel-level features into a smaller set of "super-pixels;" to do so, they ran an off-the-shelf segmentation algorithm that identified regions in the input image and varied the color of some these regions when generating "similar" images. While LIME provides no formal guarantees about its explanations, studies showed that LIME's explanations helped users evaluate which of several classifiers best generalizes.

Choice of Explanatory Vocabulary: Ribeiro *et al.*'s use of image classification decisions to explain pre-segmented image regions illustrates the larger problem of determining an explanatory vocabulary. Clearly, it would not make sense to try to identify the exact pixel that led to the decision: pixels are too low level a representation and are not semantically meaningful to users. In fact, deep neural network's power comes from the very fact that their hidden layers are trained to recognize latent features in a manner that seems to perform much better than previous efforts to define

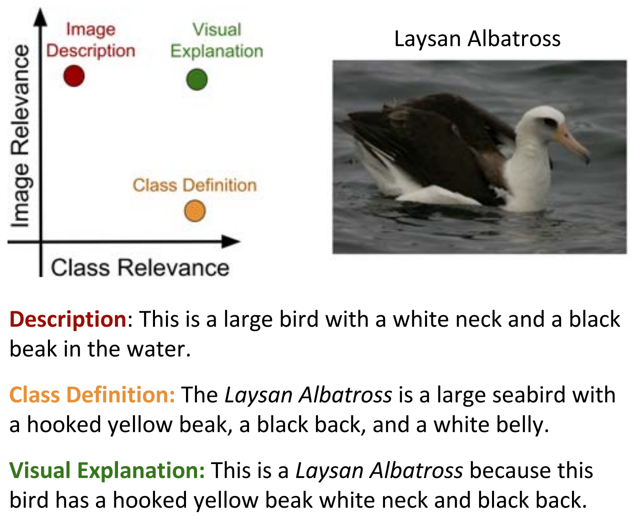

Figure 6: A visual explanation taken from [13]: "Visual explanations are both image relevant and class relevant. In contrast, image descriptions are image relevant, but not necessarily class relevant, and class definitions are class relevant but not necessarily image relevant."

such features independently. Deep networks are inscrutable exactly because we do not know what those hidden features denote.

To explain the behavior of such models, however, we must find some high-level abstraction over the input pixels that communicates the model's essence. Ribeiro *et al.*'s decision to use an off-the-shelf image-segmentation system was pragmatic. The regions it selected are easily visualized and carry some semantic value. However, regions are chosen without any regard to how the classifier makes a decision. To explain a blackbox model, where there is no possible access to the classifier's internal representation, there is likely no better option; any explanation will lack faithfulness.

However, if a user can access the classifier and tailor the explanation system to it, there are ways to choose a more meaningful vocabulary. One interesting method jointly trains a classifier with a natural language, image captioning system [13]. The classifier uses training data that is labeled with the objects appearing in the image; the captioning system is labeled with English sentences describing the appearance of the image. By training these systems jointly, the variables in the hidden layers may get aligned to semantically meaningful concepts even as they are being trained to provide discriminative power. This results in English language descriptions of images that have both high image relevance (from the captioning training data) and high class relevance (from the object recognition training data), as shown in Figure 6.

While this method works well for many examples, some explanations include details that are not actually present in the image; newer approaches, such as phrase-critic methods, may create even better descriptions [14]. Another approach might determine if there are hidden layers in the learned classifier that learn concepts corresponding to something meaningful. For example, Zeiler and Fergus observed that certain layers may function as edge or pattern detectors [40]. Whenever a user can identify the presence of such layers, then it may be preferable to use them in the explanation. Bau *et al.* describe an automatic mechanism for matching CNN representations with semantically meaningful concepts using a large, labeled corpus of objects, parts, and texture; furthermore, using this alignment, their method quantitatively scores CNN interpretability, potentially suggesting a way to optimize for intelligible models.

However, many obstacles remain. As one example, it is not clear that there *are* satisfying ways to describe important, discriminative features, which are often intangible, *e.g.*, textures. An intelligible explanation may need to define new terms or combine language with other modalities, like patches of an image. Another challenge is inducing first-order, relational descriptions, which would enable descriptions such as "a spider because it has eight legs" and "full because all seats are occupied." While quantified and relational abstractions are very natural for people, progress in statistical-relational learning has been slow and there are many open questions for neuro-symbolic learning [3].

Mapping Control Actions: Generating an explanation by mapping an inscrutable model into a simpler, explanatory model is only half of the battle. In addition to answering counterfactuals about the original model, we would ideally be able to map any control actions the user takes in the explanatory model back as adjustments to the original, inscrutable model. For example, Section 3 illustrated how a user could directly edit a GA^{2}M's shape curve (Figure 4(b)) to change the model's response to asthma. Is there a way to interpret such an action, made to an intelligible explanatory model, as a modification to the original, inscrutable model? It seems unlikely that we'll discover a general method to do this for arbitrary source models, since the abstraction mapping is not invertible in general. However, there are likely methods for mapping backwards to specific classes of source models or for specific types of feature-transform mappings. This is an important area for future study.

5 TOWARDS INTERACTIVE EXPLANATION

The optimal choice of explanation depends on the audience. Just as a human teacher would explain physics differently to students who know or do not yet know calculus, the technical sophistication and background knowledge of the recipient affects the suitability of a machine-generated explanation. Furthermore, the concerns of a house seeker whose mortgage application was denied due to a FICO score differ from those of a developer or data scientist debugging the system. Therefore, an ideal explainer should model the user's background over the course of many interactions.

The HCI community has long studied mental models [31], and many intelligent tutoring systems (ITSs) build explicit models of students' knowledge and misconceptions [2]. However, the framework for these models are typically hand-engineered for each subject domain, so it may be difficult to adapt ITS approaches to a system that aims to explain an arbitrary black-box learner.

Even with an accurate user model, it is likely that an explanation will not answer all of a user's concerns, because the human may have follow-up questions. We conclude that an explanation system should be *interactive*, supporting such questions from and actions by the user. This matches results from psychology literature, summarized in Section 2, and highlights Grice's maxims, especially

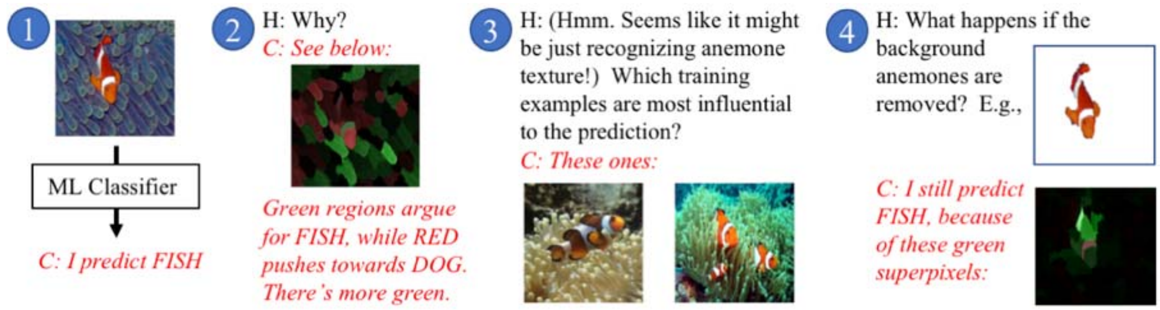

Figure 7: An example of an interactive explanatory dialog for gaining insight into a DOG/FISH image classifier. (For illustration, the questions and answers are shown in English language text, but our use of a 'dialog' is for illustration only. An interactive GUI, e.g., building on the ideas of Krause et al. [20], would likely be a better realization.)

those pertaining to quantity and relation. It also builds on Lim and Dey's work in ubiquitous computing, which investigated the kinds of questions users wished to ask about complex, context-aware applications [24]. We envision an interactive explanation system that supports many different follow-up and drill-down action after presenting a user with an initial explanation:

- *Redirecting the answer by changing the foil*: "Sure, but why didn't you predict class C?"
- *Asking for more detail* (*i.e.*, a more complex explanatory model), perhaps while restricting the explanation to a subregion of feature space: "I'm only concerned about women over age 50..."
- *Asking for a decision's rationale*: "What made you believe this?" To which the system might respond by displaying the labeled *training examples* that were most influential in reaching that decision, *e.g.*, ones identified by influence functions [19] or nearest neighbor methods.
- *Query the model's sensitivity* by asking what minimal perturbations to certain features would lead to a different output.
- *Changing the vocabulary* by adding (or removing) a feature in the explanatory model, either from a predefined set, by using methods from machine teaching, or with concept activation vectors [17].
- *Perturbing the input example* to see the effect on both prediction and explanation. In addition to aiding understanding of the model (directly testing a counterfactual), this action enables an affected user who wants to contest the initial prediction: "But officer, one of those prior DUIs was overturned...?"
- *Adjusting the model:* Based on new understanding, the user may wish to correct the model. Here, we expect to build on tools for interactive machine learning [1] and explanatory debugging [20, 21], which have explored interactions for adding new training examples, correcting erroneous labels in existing data, specifying new features, and modifying shape functions. As mentioned in the previous section, it may be challenging to map user adjustments, that are made in reference to an explanatory model, back into the original, inscrutable model.

To make these ideas concrete, Figure 7 presents a possible dialog as a user tries to understand the robustness of a deep neural dog/fish classifier built atop Inception v3 [39]. As the figure shows: 1) The computer correctly predicts that the image depicts a fish. 2) The user requests an explanation, which is provided using LIME [33]. 3) The user, concerned that the classifier is paying more attention to the background than to the fish itself, asks to see the training data that influenced the classifier; the nearest neighbors are computed using influence functions [19]. While there are anemones in those images, it also seems that the system is recognizing a clownfish. 4) To gain confidence, the user edits the input image to remove the background, resubmits it to the classifier and checks the explanation.

6 EXPLAINING COMBINATORIAL SEARCH

Most of the preceding discussion has focused on intelligible *machine learning*, which is just one type of artificial intelligence. However, the same issues also confront systems based on *deep-lookahead search*. While many search algorithms have strong theoretical properties, true correctness depends on assumptions made when modeling the underlying actions [29] that enable a user to question the agent's choices.

Consider a planning algorithm that has generated a sequence of actions for a remote, mobile robot. If the plan is short with a moderate number of actions, then the problem may be inherently intelligible, but larger search spaces will likely be cognitively overwhelming. In these cases, local explanations offer a simplification technique that is helpful, just as it was when explaining machine learning. The vocabulary issue is likewise crucial: how does one succinctly summarize a complete search subtree abstractly? Depending on the choice of explanatory foil, different answers are appropriate [8]. Sreedharan et al. describe an algorithm for generating the minimal explanation that patches a user's partial understanding of a domain [37]. Work on mixed-initiative planning [7] has demonstrated the importance of supporting interactive dialog with a planning system.

Many AI systems, such as AlphaGo [35], combine both deep search *and* machine learning; these will be especially hard to explain since complexity arises from the interaction of combinatorics and a learned model.

7 FINAL THOUGHTS

In order to trust deployed AI systems, we must not only improve their robustness [5], but also develop ways to make their reasoning intelligible. Intelligibility will help us spot AI that makes mistakes due to distributional drift or incomplete representations of goals and features. Intelligibility will also facilitate control by humans in increasingly common collaborative human/AI teams. Furthermore, intelligibility will help humans learn from AI. Finally, there are legal reasons to want intelligible AI, including the European GDPR and a growing need to assign liability when AI errs.

Depending on the complexity of the models involved, two approaches to enhancing understanding may be appropriate: 1) using an inherently interpretable model, or 2) adopting an inscrutably complex model and generating post hoc explanations by mapping it to a simpler, explanatory model through a combination of currying and local approximation. When learning a model over a medium number of human-interpretable features, one may confidently balance performance and intelligibility with approaches like GA^2Ms. However, for problems with thousands or millions of features, performance requirements likely force the adoption of inscrutable methods, such as deep neural networks or boosted decision trees. In these situations, post-hoc explanations may be the only way to facilitate human understanding.

Research on explanation algorithms is developing rapidly, with work on both local (instance-specific) explanations and global approximations to the learned model. A key challenge for all these approaches is the construction of an explanation vocabulary, essentially a set of features used in the approximate explanation model. Different explanatory models may be appropriate for different choices of explanatory foil, an aspect deserving more attention from systems builders. While many intelligible models can be directly edited by a user, more research is needed to determine how best to map such actions back to modify an underlying inscrutable model. Results from psychology show that explanation is a social process, best thought of as a conversation. As a result, we advocate increased work on *interactive* explanation systems that support a wide range of follow-up actions. To spur rapid progress in this important field, we hope to see collaboration between researchers in multiple disciplines.

Acknowledgements: We thank E. Adar, S. Ameshi, R. Calo, R. Caruana, M. Chickering, O. Etzioni, J. Heer, E. Horvitz, T. Hwang, R. Kambhamapti, E. Kamar, S. Kaplan, B. Kim, P. Simard, Mausam, C. Meek, M. Michelson, S. Minton, B. Nushi, G. Ramos, M. Ribeiro, M. Richardson, P. Simard, J. Suh, J. Teevan, and T. Wu for helpful conversations and comments. This work was supported in part by the Future of Life Institute grant 2015-144577 (5388) with additional support from NSF grant IIS-1420667, ONR grant N00014-15-1-2774, and the WRF/Cable Professorship.

REFERENCES

[1] S. Amershi, M. Cakmak, W. Knox, and T. Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Magazine* 35, 4 (2014), 105–120.

[2] J. R Anderson, F. Boyle, and B. Reiser. 1985. Intelligent tutoring systems. *Science* 228, 4698 (1985), 456–462.

[3] T. Besold, A. d'Avila Garcez, S. Bader, H. Bowman, P. Domingos, P. Hitzler, K. Kühnberger, L. Lamb, D. Lowd, P. Lima, L. de Penning, G. Pinkas, H. Poon, and G. Zaverucha. 2017. Neural-Symbolic Learning and Reasoning: A Survey and Interpretation. *CoRR* abs/1711.03902 (2017). arXiv:1711.03902

[4] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *KDD*.

[5] T. Dietterich. 2017. Steps Towards Robust Artificial Intelligence. *AI Magazine* 38, 3 (2017).

[6] F. Doshi-Velez and B. Kim. 2017. Towards A Rigorous Science of Interpretable Machine Learning. *ArXiv* (2017). arXiv:1702.08608

[7] George Ferguson and James F. Allen. 1998. TRIPS: An Integrated Intelligent Problem-Solving Assistant. In *AAAI/IAAI*.

[8] M. Fox, D. Long, and D. Magazzeni. 2017. Explainable Planning. http://arxiv.org/abs/1709.10256

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy. 2014. Explaining and Harnessing Adversarial Examples. *ArXiv* (2014). arXiv:1412.6572

[10] P. Grice. 1975. *Logic and Conversation.* 41–58.

[11] J. Halpern and J. Pearl. 2005. Causes and explanations: A structural-model approach. Part I: Causes. *The British journal for the philosophy of science* 56, 4 (2005), 843–887.

[12] M. Hardt, E. Price, and N. Srebro. 2016. Equality of opportunity in supervised learning. In *NIPS*.

[13] L. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell. 2016. Generating visual explanations. In *ECCV*.

[14] L. A. Hendricks, R. Hu, T. Darrell, and Z. Akata. 2017. Grounding Visual Explanations. *ArXiv* (2017). arXiv:1711.06465

[15] D. Hilton. 1990. Conversational processes and causal explanation. *Psychological Bulletin* 107, 1 (1990), 65.

[16] D. Kahneman. 2011. *Thinking, fast and slow.* Farrar, Straus and Giroux, New York. http://a.co/hGYmXGJ

[17] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres. 2017. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). *ArXiv e-prints* (Nov. 2017). arXiv:stat.ML/1711.11279

[18] Derek J Koehler. 1991. Explanation, imagination, and confidence in judgment. *Psychological bulletin* 110, 3 (1991), 499.

[19] P. Koh and P. Liang. 2017. Understanding black-box predictions via influence functions. In *ICML*.

[20] J. Krause, A. Dasgupta, J. Swartz, Y. Aphinyanaphongs, and E. Bertini. 2017. A Workflow for Visual Diagnostics of Binary Classifiers using Instance-Level Explanations. In *IEEE VAST*.

[21] T. Kulesza, M. Burnett, W. Wong, and S. Stumpf. 2015. Principles of explanatory debugging to personalize interactive machine learning. In *IUI*.

[22] H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec. 2017. Interpretable & Explorable Approximations of Black Box Models. *KDD-FATML* (2017).

[23] D. Lewis. 1986. Causal explanation. *Philosophical Papers* 2 (1986), 214–240.

[24] Brian Y Lim and Anind K Dey. 2009. Assessing demand for intelligibility in context-aware applications. In *Proceedings of the 11th International Conference on Ubiquitous Computing*. ACM, 195–204.

[25] Z. Lipton. 2016. The Mythos of Model Interpretability. In *ICML Workshop on Human Interpretability in ML*.

[26] T. Lombrozo. 2007. Simplicity and probability in causal explanation. *Cognitive psychology* 55, 3 (2007), 232–257.

[27] Y. Lou, R. Caruana, and J. Gehrke. 2012. Intelligible models for classification and regression. In *KDD*.

[28] S. Lundberg and S. Lee. 2017. A unified approach to interpreting model predictions. *NIPS* (2017).

[29] J. Mccarthy and P. Hayes. 1969. Some Philosophical Problems from the Standpoint of Artificial Intelligence. In *Machine Intelligence*. 463–502.

[30] T. Miller. 2017. Explanation in artificial intelligence: Insights from the social sciences. *ArXiv* (2017). arXiv:1706.07269

[31] Donald A Norman. 2014. Some Observations on Mental Models. In *Mental Models*. Psychology Press, 15–22.

[32] A. Papadimitriou, P. Symeonidis, and Y. Manolopoulos. 2012. A generalized taxonomy of explanations styles for traditional and social recommender systems. *Data Mining and Knowledge Discovery* 24, 3 (2012), 555–583.

[33] M. Ribeiro, S. Singh, and C. Guestrin. 2016. Why Should I Trust You?: Explaining the Predictions of any Classifier. In *KDD*.

[34] M. Ribeiro, S. Singh, and C. Guestrin. 2018. Anchors: High-Precision Model-Agnostic Explanations. In *AAAI*.

[35] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529, 7587 (2016), 484–489.

[36] S. Sloman. 1997. Explanatory coherence and the induction of properties. *Thinking & Reasoning* 3, 2 (1997), 81–110.

[37] S. Sreedharan, S. Srivastava, and S. Kambhampati. 2018. Hierarchical Expertise-Level Modeling for User Specific Robot-Behavior Explanations. *ArXiv e-prints* (Feb. 2018). arXiv:1802.06895

[38] W. Swartout. 1983. XPLAIN: a system for creating and explaining expert consulting programs. *Artificial Intelligence* 21, 3 (1983), 285 – 325.

[39] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *CVPR*.

[40] M. Zeiler and R. Fergus. 2014. Visualizing and understanding convolutional networks. In *ECCV*.