Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead

Nature Machine Intelligence

May 2019

Section: Pg. 206-215; Vol. 1; No. 5; ISSN: 2522-5839

Length: 9709 words

Byline: *cynthia@cs.duke.edu*

Body

Main

There has been an increasing trend in healthcare and criminal justice to leverage machine learning (ML) for high-stakes prediction applications that deeply impact human lives. Many of the ML models are black boxes that do not explain their predictions in a way that humans can understand. The lack of transparency and accountability of predictive models can have (and has already had) severe consequences; there have been cases of people incorrectly denied parole, poor bail decisions leading to the release of dangerous criminals, ML-based pollution models stating that highly polluted air was safe to breathe and generally poor use of limited valuable resources in criminal justice, medicine, energy reliability, finance and in other domains.

Rather than trying to create models that are inherently interpretable, there has been a recent explosion of work on ‘explainable ML’, where a second (post hoc) model is created to explain the first black box model. This is problematic. Explanations are often not reliable, and can be misleading, as we discuss below. If we instead use models that are inherently interpretable, they provide their own explanations, which are faithful to what the model actually computes.

In what follows, we discuss the problems with explainable ML, followed by the challenges in interpretable ML. This document is mainly relevant to high-stakes decision making and troubleshooting models, which are the main two reasons one might require an interpretable or explainable model. Interpretability is a domain-specific notion–, so there cannot be an all-purpose definition. Usually, however, an interpretable machine learning model is constrained in model form so that it is either useful to someone, or obeys structural knowledge of the domain, such as monotonicity (for example, ref.), causality, structural (generative) constraints, additivity or physical constraints that come from domain knowledge. Interpretable models could use case-based reasoning for complex domains. Often for structured data, sparsity is a useful measure of interpretability, because humans can handle at most 7 ± 2 cognitive entities at once,. Sparse models allow a view of how variables interact jointly rather than individually. We will discuss several forms of interpretable machine ML models for different applications, but there can never be a single definition; for example, in some domains sparsity is useful, and in others it is not. There is a spectrum between fully transparent models (where we understand how all the variables are jointly related to each other) and models that are lightly constrained in model form (such as models that are forced to increase as one of the variables increases, or models that, all else being equal, prefer variables that domain experts have identified as important; see ref.).

A preliminary version of this manuscript appeared at a workshop, entitled ‘Please stop explaining black box machine learning models for high stakes decisions’.

Key issues with explainable ML

A black box model could be either (1) a function that is too complicated for any human to comprehend or (2) a function that is proprietary (Supplementary Section). Deep learning models, for instance, tend to be black boxes of the first kind because they are highly recursive. As the term is presently used in its most common form, an explanation is a separate model that is supposed to replicate most of the behaviour of a black box (for example, 'the black box says that people who have been delinquent on current credit are more likely to default on a new loan'). Note that the term 'explanation' here refers to an understanding of how a model works, as opposed to an explanation of how the world works. The terminology 'explanation' will be discussed later; it is misleading. I am concerned that the field of interpretability/explainability/comprehensibility/transparency in ML has strayed away from the needs of real problems. This field dates back to the early 1990s at least (see refs. ,), and there are a huge number of papers on interpretable ML in various fields (that often do not have the word 'interpretable' or 'explainable' in the title, as recent papers do). Recent work on the explainability of black boxes—rather than the interpretability of models—contains and perpetuates critical misconceptions that have generally gone unnoticed, but that can have a lasting negative impact on the widespread use of ML models in society. Let us spend some time discussing this before discussing possible solutions.

It is a myth that there is necessarily a trade-off between accuracy and interpretability

There is a widespread belief that more complex models are more accurate, meaning that a complicated black box is necessary for top predictive performance. However, this is often not true, particularly when the data are structured, with a good representation in terms of naturally meaningful features. When considering problems that have structured data with meaningful features, there is often no significant difference in performance between more complex classifiers (deep neural networks, boosted decision trees, random forests) and much simpler classifiers (logistic regression, decision lists) after preprocessing. (Supplementary Section discusses this further.) In data science problems, where structured data with meaningful features are constructed as part of the data science process, there tends to be little difference between algorithms, assuming that the data scientist follows a standard process for knowledge discovery (such as KDD, CRISP-DM or BigData; see refs. –).

Even for applications such as computer vision, where deep learning has major performance gains, and where interpretability is much more difficult to define, some forms of interpretability can be imbued directly into the models without losing accuracy. This will be discussed more later in the section ''. Uninterpretable algorithms can still be useful in high-stakes decisions as part of the knowledge discovery process, for instance to obtain baseline levels of performance, but they are not generally the final goal of knowledge discovery.

Figure , taken from the DARPA Explainable Artificial Intelligence program's Broad Agency Announcement, exemplifies a blind belief in the myth of the accuracy–interpretability trade-off. This is not a 'real' figure, in that it was not generated by any data. The axes have no quantification (there is no specific meaning to the horizontal or vertical axes). The image appears to illustrate an experiment with a static data set, where several ML algorithms are applied to the same data set. However, this kind of smooth accuracy/interpretability/explainability trade-off is atypical in data science applications with meaningful features. Even if one were to quantify the interpretability/explainability axis and aim to show that such a trade-off did exist, it is not clear what algorithms would be applied to produce this figure. (Would one actually claim it is fair to compare the 1984 decision tree algorithm CART to a 2018 deep learning model and conclude that interpretable models are not as accurate?) One can always create an artificial trade-off between accuracy and interpretability/explainability by removing parts of a more complex model to reduce accuracy, but this is not representative of the analysis one would perform on a real problem. It is also not clear why the comparison should be performed on a static data set, because any formal process for defining knowledge from data– would require an iterative process, where one refines the data processing after interpreting the results. Generally, in the practice of data science, the small difference in performance between ML algorithms can be overwhelmed by the ability to interpret results and process the data better at the next iteration. In those cases, the accuracy/interpretability trade-off is reversed—more interpretability leads to better overall accuracy, not worse.

A fictional depiction of the accuracy–interpretability trade-off.

Adapted from ref. , DARPA.

Efforts working within a knowledge discovery process led me to work in interpretable ML. Specifically, I participated in a large-scale effort to predict electrical grid failures across New York City. The data were messy, including free text documents (trouble tickets), accounting data about electrical cables from as far back as the 1890s and inspections data from a brand new manhole inspections programme; even the structured data were not easily integrated into a database, and there were confounding issues and other problems. Algorithms on a static data set were at most 1% different in performance, but the ability to interpret and reprocess the data led to significant improvements in performance, including correcting problems with the data set and revealing false assumptions about the data generation process. The most accurate predictors we found were sparse models with meaningful features that were constructed through the iterative process.

The belief that there is always a trade-off between accuracy and interpretability has led many researchers to forgo the attempt to produce an interpretable model. This problem is compounded by the fact that researchers are now trained in deep learning, but not in interpretable ML. Worse, toolkits of ML algorithms offer little in the way of useful interfaces for interpretable ML methods.

To our knowledge, all recent review and commentary articles on this topic imply (implicitly or explicitly) that the trade-off between interpretability and accuracy generally occurs. It could be possible that there are application domains where a complete black box is required for a high stakes decision. As of yet, I have not encountered such an application, despite having worked on numerous applications in healthcare and criminal justice (for example, ref.), energy reliability (for example, ref.) and financial risk assessment (for example, ref.).

Explainable ML methods provide explanations that are not faithful to what the original model computes

Explanations must be wrong. They cannot have perfect fidelity with respect to the original model. If the explanation was completely faithful to what the original model computes, the explanation would equal the original model, and one would not need the original model in the first place, only the explanation. (In other words, this is a case where the original model would be interpretable.) This leads to the danger that any explanation method for a black box model can be an inaccurate representation of the original model in parts of the feature space. (See also ref. , among others.)

An inaccurate (low-fidelity) explanation model limits trust in the explanation, and by extension, trust in the black box that it is trying to explain. An explainable model that has a 90% agreement with the original model indeed explains the original model most of the time. However, an explanation model that is correct 90% of the time is wrong 10% of the time. If a tenth of the explanations are incorrect, one cannot trust the explanations, and thus one cannot trust the original black box. If we cannot know for certain whether our explanation is correct, we cannot know whether to trust either the explanation or the original model.

A more important misconception about explanations stems from the terminology ‘explanation’, which is often used in a misleading way, because explanation models do not always attempt to mimic the calculations made by the original model. Even an explanation model that predicts almost identically to a black box model might use completely different features, and is thus not faithful to the computation of the black box. Consider a black box model for criminal recidivism prediction, where the goal is to predict whether someone will be arrested within a certain time after being released from jail/prison. Most recidivism prediction models depend explicitly on age and criminal history, but do not explicitly depend on race. Because criminal history and age are correlated with race in all of our data sets, a fairly accurate explanation model could construct a rule such as ‘This person is predicted to be arrested because they are black’. This might be an accurate explanation model because it correctly mimics the predictions of the original model, but it would not be faithful to what the original model computes. This is possibly the main flaw identified by criminologists in the ProPublica analysis, that accused the proprietary COMPAS recidivism model of being racially biased. COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) is a proprietary model that is used widely in the US justice system for parole and bail decisions. ProPublica created a linear explanation model for COMPAS that depended on race, and then accused the black

box COMPAS model of depending on race, conditioned on age and criminal history. In fact, COMPAS seems to be nonlinear, and it is entirely possible that COMPAS does not depend on race (beyond its correlations with age and criminal history). ProPublica's linear model was not truly an 'explanation' for COMPAS, and they should not have concluded that their explanation model uses the same important features as the black box it was approximating. (There will be a lot more discussion about COMPAS later in this document.)

An easy fix to this problem is to change terminology. Let us stop calling approximations to black box model predictions 'explanations'. For a model that does not use race explicitly, an automated explanation 'This model predicts you will be arrested because you are black' is not an explanation of what the model is actually doing, and would be confusing to a judge, lawyer or defendant. It can be much easier to detect and debate possible bias or unfairness with an interpretable model than with a black box. Similarly, it could be easier to detect and avoid data privacy issues with interpretable models than black boxes. Just as in the recidivism example above, many of the methods that claim to produce explanations instead compute useful summary statistics of predictions made by the original model. Rather than producing explanations that are faithful to the original model, they show trends in how predictions are related to the features. Calling these 'summaries of predictions', 'summary statistics' or 'trends' rather than 'explanations' would be less misleading.

Explanations often do not make sense or do not provide enough detail to understand what the black box is doing

Even if both models are correct (the original black box is correct in its prediction and the explanation model is correct in its approximation of the black box's prediction), it is possible that the explanation leaves out so much information that it makes no sense. I will give an example from image processing, for a low-stakes decision (not a high-stakes decision where explanations are needed, but where explanation methods are often demonstrated). Saliency maps are often considered to be explanatory. Saliency maps can be useful to determine what part of the image is being omitted by the classifier, but this leaves out all information about how relevant information is being used. Knowing where the network is looking within the image does not tell the user what it is doing with that part of the image, as illustrated in Fig. . In fact, the saliency maps for multiple classes could be essentially the same; in that case, the explanation for why the image might contain a Siberian husky would be the same as the explanation for why the image might contain a transverse flute.

Saliency does not explain anything except where the network is looking.

We have no idea why this image is labelled as either a dog or a musical instrument when considering only saliency. The explanations look essentially the same for both classes. Credit: Chaofen Chen, Duke University

An unfortunate trend in recent work is to show explanations only for the observation's correct label when demonstrating the method (for example, Fig. would not appear). Demonstrating a method using explanations only for the correct class is misleading. This practice can instil a false sense of confidence in the explanation method and in the black box. Consider, for instance, a case where the explanations for multiple (or all) of the classes are identical. This situation would happen often when saliency maps are the explanations, because they tend to highlight edges, and thus provide similar explanations for each class. These explanations could be identical even if the model is always wrong. Then, showing only the explanations for the image's correct class misleads the user into thinking that the explanation is useful, and that the black box is useful, even if neither one of them is.

Saliency maps are only one example of explanations that are so incomplete that they might not convey why the black box predicted what it did. Similar arguments can be made with other kinds of explanation methods. Poor explanations can make it very hard to troubleshoot a black box.

Black box models are often not compatible with situations where information outside the database needs to be combined with a risk assessment

In high-stakes decisions there are often considerations outside the database that need to be combined with a risk calculation. For instance, what if the circumstances of the crime are much worse than a generic assigned charge? There are often circumstances whose knowledge could either increase or decrease someone's risk. But if the

model is a black box, it is very difficult to manually calibrate how much this additional information should raise or lower the estimated risk. This issue arises constantly; for instance, the proprietary COMPAS model used in the US justice system for recidivism risk prediction does not depend on the seriousness of the current crime,. Instead, the judge is instructed to somehow manually combine current crime with COMPAS. Actually, it is possible that many judges do not know this fact. If the model were transparent, the judge could see directly that the seriousness of the current crime is not being considered in the risk assessment.

Black box models with explanations can lead to an overly complicated decision pathway that is ripe for human error

Typographical errors seem to be common in computing COMPAS, and these typographical errors sometimes determine bail decision outcomes,. This exemplifies an important drawback of using overly complicated black box models for recidivism prediction—they may be incorrectly calculated in practice. The computation of COMPAS requires 130+ factors. If typographical errors by humans entering these data into a survey occur at a rate of 1%, then more than one out of every two surveys on average will have at least one typographical error. The multitude of typographical errors has been argued to be a type of procedural unfairness, whereby two individuals who are identical might be randomly given different parole or bail decisions. These types of error have the potential to reduce the in-practice accuracy of these complicated models.

On the separate topic of model troubleshooting, an overly complicated black box model may be flawed but we do not know it, because it is difficult to troubleshoot. Having an (incomplete) explanation of it may not help, and now we must troubleshoot two models rather than one (the black box model and the explanation model).

In the next section, we completely switch gears. We will discuss reasons why so many people appear to advocate for black box models with separate explanation models, rather than inherently interpretable models—even for high-stakes decisions.

Key issues with interpretable ML

There are many cases where black boxes with explanations are preferred over interpretable models, even for high-stakes decisions. However, for most applications, I am hopeful that there are ways around some of these problems, whether they are computational problems or problems with training of researchers and availability of code. The first problem, however, is currently a major obstacle that I see no way of avoiding other than through policy, as discussed in the next section.

Corporations can make profits from the intellectual property afforded to a black box

Companies that charge for individual predictions could find their profits obliterated if an interpretable model were used instead.

Consider the COMPAS proprietary recidivism risk prediction tool discussed above that is in widespread use in the US justice system for predicting the probability that someone will be arrested after their release.

The COMPAS model is equally accurate for recidivism prediction as the very simple three-rule interpretable ML model involving only age and number of past crimes shown in Table . However, there is no clear business model that would suggest profiting from the simple transparent model. The simple model in Table was created from an algorithm called Certifiably Optimal Rule Lists (CORELS) that looks for if–then patterns in data. Even though the model in Table looks like a rule of thumb that a human may have designed without data, it is instead a full-blown ML model. A qualitative comparison of the COMPAS and CORELS models is in Table . Standard ML tools and interpretable ML tools seem to be approximately equally accurate for predicting recidivism, even if we define recidivism in many different ways, for many different crime types,. This evidence, however, has not changed the momentum of the justice system towards proprietary models. As of writing, California has recently eliminated its cash bail system, instead enforcing that decisions be made by algorithms; it is unclear whether COMPAS will be the algorithm used for this, despite the fact that it is not known to be any more accurate than other models, such as the simple CORELS model in Table .

Machine learning model from the CORELS algorithm

IF	age between 18?20 and sex is male	THEN predict arrest (within 2 years)
ELSE IF	age between 21?23 and 2?3 prior offences	THEN predict arrest
ELSE IF	more than three priors	THEN predict arrest
ELSE	predict no arrest	

This model from ref. is the minimizer of a special case of equation () discussed later in the challenges section. CORELS' code is open source and publicly available at *http://corels.eecs.harvard.edu/*, along with the data from Florida needed to produce this model.

Comparison of COMPAS and CORELS models

COMPAS	**CORELS**
Black box; 130+ factors; might include socio-economic info; expensive (software licence); within software used in US justice system	Full model is in Table ; only age, priors, gender (optional); no other information; free, transparent

COMPAS is not a ML model—it was not created by any standard ML algorithm. It was designed by experts based on carefully designed surveys and expertise, and it does not seem to depend heavily on past criminal history. Interestingly, if the COMPAS model were not proprietary, its documentation indicates that it would actually be an interpretable predictive model. (It is a black box of the second type—proprietary—but not the first type—complicated—discussed above.) Revealing this model, however, would be revealing a trade secret.

Let us switch examples to consider the proprietary ML model by BreezoMeter, used by Google during the California wildfires of 2018, which predicted air quality as 'good—ideal air quality for outdoor activities', when air quality was dangerously bad according to multiple other models and people reported their cars covered in ash. The Environmental Protection Agency's free, vigorously tested air quality index would have provided a reliable result. How could BreezoMeter's ML method be so badly wrong and put so many in danger? We will never find out, but BreezoMeter, who has probably made a profit from making these predictions, may not have developed this new technology if its models were forced to be transparent.

In medicine, there is a trend towards blind acceptance of black box models, which will open the door for companies to sell more models to hospitals. For example, radiology and in-hospital patient monitoring are areas of medicine that stand to gain tremendously by automation; humans cannot process data fast enough or rapidly enough to compete with machines. However, in trusting these automated systems, we must also trust the full database on which they were trained, the processing of the data, along with the completeness of the database. If the database does not represent the full set of possible situations that can arise, then the model could be making predictions in cases that are very different from anything on which it was trained. An example of where this can go wrong is given by Zech and colleagues, who noticed that their neural network was picking up on the word 'portable' within an X-ray image, representing the type of X-ray equipment rather than the medical content of the image. If they had used an interpretable model, or even an explainable model, this issue would never have gone unnoticed. The issue of counfounding generally is highlighted in ref. . In fact, the plague of confounding haunts a vast number of data sets, and particularly medical data sets. This means that proprietary models for medicine can have serious errors. These models can also be fragile, in that if the model is used in practice in a slightly different setting than how it was trained (for example, new X-ray equipment), accuracy can drop substantially.

The examples of COMPAS, Breezometer and black box medical diagnosis all illustrate a problem with the business model for ML. In particular, there is a conflict of responsibility in the use of black box models for high-stakes decisions: the companies that profit from these models are not necessarily responsible for the quality of individual predictions. A prisoner serving an excessively long sentence due to a mistake entered in an overly complicated risk score could suffer for years, whereas the company that constructed this complicated model is unaffected. On the contrary, the fact that the model was complicated and proprietary allowed the company to profit from it. In that

sense, a model's designers are not incentivized to be careful in its design, performance and ease of use. These are some of the same types of problems affecting the credit rating agencies who priced mortgages in 2008—indeed these are the same problems that contributed to the financial crisis in the United States at that time.

One argument favouring black boxes is that keeping these models hidden prevents them from being gamed or reverse-engineered. It is not clear that this argument generally makes sense. In fact, the reason a system may be gamed is because it most likely was not designed properly in the first place, leading to a form of Goodhart's law if it were revealed. Quoting from ref. about product rating systems: "If the ratings are accurate measures of quality, then making the ratings more transparent could have a uniformly positive impact: it would help companies to make better rated products, it would help consumers to have these higher quality products, and it would encourage rating companies to receive feedback as to whether their rating systems fairly represent quality." Thus, transparency could help improve the quality of the system, whereby attempting to game it would genuinely align with the overall goal of improvement. For example, improving one's credit score should actually correspond to an improvement in creditworthiness.

Another argument favouring black boxes is the belief that 'counterfactual explanations' of black boxes are sufficient. A counterfactual explanation describes a minimal change to the input that would result in the opposite prediction. For instance, a possible counterfactual explanation might be 'your loan application was denied, but if you had $1,000 less debt, you would have qualified for the loan'. This type of explanation can suffer from the key issue already discussed above, regarding combining information outside the database with the black box (see the subsection 'Black box models are often not compatible with situations where information outside the database needs to be combined with a risk assessment'). In particular, the 'minimal' change to the input might be different for different individuals. Supplementary Section discusses in more depth why counterfactual explanations generally do not suffice for high-stakes decisions of black boxes.

Interpretable models can entail significant effort to construct in terms of both computation and domain expertise

As discussed above, interpretability usually translates in practice to a set of application-specific constraints on the model. Solving constrained problems is generally harder than solving unconstrained problems. Domain expertise is needed to construct the definition of interpretability for the domain, and the features for ML. For data that are unconfounded, complete and clean, it is much easier to use a black box ML method than to troubleshoot and solve computationally hard problems. However, for high-stakes decisions, analyst time and computational time are less expensive than the cost of having a flawed or overly complicated model. That is, it is worthwhile to devote extra effort and cost into constructing a high-quality model. Even so, many organizations do not have analysts who have the training or expertise to construct interpretable models at all.

Some companies have started to provide interpretable ML solutions using proprietary software. Although this is a step in the right direction, it is not clear that the proprietary software is better than publicly available software. For example, claims made by some companies about performance of their proprietary algorithms are not impressive (for example, Interpretable AI, whose decision tree performance using mixed integer programming software in 2017 is reported to be often beaten by or comparable to the 1984 Classification and Regression Tree algorithm, CART).

As discussed earlier, interpretability constraints (like sparsity) lead to optimization problems that have been proven to be computationally hard in the worst case. The theoretical hardness of these problems does not mean we cannot solve them, although in real cases these optimization problems are often difficult to solve. Major improvements have been made in the past decade, and some are discussed later in the '' section. Explanation methods, on the other hand, are usually based on simple derivatives, which lead to easier gradient-based optimization.

Black box models seem to uncover 'hidden patterns'

The fact that many scientists have difficulty constructing interpretable models may be fueling the belief that black boxes have the ability to uncover subtle hidden patterns in the data about which the user was not previously aware. A transparent model may be able to uncover these same patterns. If the pattern in the data was important enough that a black box model could leverage it to obtain better predictions, an interpretable model might also locate the

same pattern and use it. Again, this depends on the ML researcher's ability to create accurate yet interpretable models. The researcher needs to create a model that has the capability of uncovering the types of pattern that the user would find interpretable, but also the model needs to be flexible enough to fit the data accurately. This, and the optimization challenges discussed above, are where the difficulty lies with constructing interpretable models.

Encouraging responsible ML governance

Currently, the European Union's revolutionary General Data Protection Regulation and other AI regulation plans govern 'right to an explanation', where only an explanation is required, not an interpretable model, in particular 'The data subject shall have the right not to be subject to a decision based solely on automated processing, including profiling, which produces legal effects concerning him or her or similarly significantly affects him or her' (Article 22 of GDPR regulations from *http://www.privacy-regulation.eu/en/22.htm*). If one were to provide an explanation for an automated decision, it is not clear whether the explanation is required to be accurate, complete or faithful to the underlying model (for example, see ref.). Less than satisfactory explanations can easily undermine these new policies.

Let us consider a possible mandate that, for certain high-stakes decisions, no black box should be deployed when there exists an interpretable model with the same level of performance. If such a mandate were deployed, organizations that produce and sell black box models could then be held accountable if an equally accurate transparent model exists. It could be considered a form of false advertising to sell a black box model if there is an equally accurate interpretable model. The onus would then fall on organizations to produce black box models only when no transparent model exists for the same task.

This possible mandate could produce a change in the business model for ML. Opacity is viewed as essential in protecting intellectual property, but it is at odds with the requirements of many domains that involve public health or welfare. However, the combination of opacity and explainability is not the only way to incentivize ML experts to invest in creating such systems. Compensation for developing an interpretable model could be provided in a lump sum, and the model could be released to the public. The creator of the model would not be able to profit from licensing the model over a period of time, but the fact that the models are useful for public good applications would make these problems appeal to academics and charitable foundations.

This proposal will not solve all problems, but it could at least rule out companies selling recidivism prediction models, possibly credit scoring models and other kinds of models where we can construct accurate yet interpretable alternatives. If applied too broadly, it could reduce industrial participation in cases where machine learning might benefit society.

Consider a second proposal, which is weaker than the one provided above, but which might have a similar effect. Let us consider the possibility that organizations that introduce black box models would be mandated to report the accuracy of interpretable modelling methods. In that case, one could more easily determine whether the accuracy/interpretability trade-off claimed by the organization is worthwhile. This also forces the organization to try using interpretable modelling methods. It also encourages the organization to use these methods carefully, otherwise risking the possibility of criticism.

As mentioned earlier, I have not yet found a high-stakes application where a fully black box model is necessary, despite having worked on many applications. As long as we continue to allow for a broad definition of interpretability that is adapted to the domain, we should be able to improve decision making for serious tasks of societal importance. However, in order for people to design interpretable models, the technology must exist to do so. As discussed earlier, there is a formidable computational hurdle in designing interpretable models, even for standard structured data with already meaningful features.

Algorithmic challenges in interpretable ML

What if every black box ML model could be replaced with one that was equally accurate but also interpretable? If we could do this, we would identify flaws in our models and data that we could not see before. Perhaps we could

prevent some of the poor decisions in criminal justice and medicine that are caused by problems with using black box models. We could also eliminate the need for explanations that are misleading and often wrong.

Because interpretability is domain-specific, a large toolbox of possible techniques can come in handy. Below we expand on three of the challenges for interpretable ML that appear often. All three cases have something in common: people have been providing interpretable predictive models for these problems for decades, and the human-designed models look just like the type of model we want to create with ML. I also discuss some of our current work on these well-known problems.

Each of these challenges is a representative from a major class of models: modelling that uses logical conditions (Challenge 1), linear modelling (Challenge 2) and case-based reasoning (Challenge 3). By no means is this set of challenges close to encompassing the large number of domain-specific challenges that exist in creating interpretable models.

Challenge 1: constructing optimal logical models

A logical model consists of statements involving ‘or’, ‘and’, ‘if–then’ and so on. The CORELS model in Table is a logical model, called a rule list. Decision trees are logical models, as well as conjunctions of disjunctions (‘or’s of ‘and’s—for instance, IF condition A is true OR conditions B AND C are true, THEN predict yes, otherwise predict no).

Logical models have been crafted by hand as expert systems as far back as the 1970s. Since then, there have been many heuristics for creating logical models; for example, one might add logical conditions one by one (greedily), and then prune conditions away that are not helpful (again, greedily). These heuristic methods tend to be inaccurate and/or uninterpretable because they do not choose a globally best choice (or approximately best choice) for the logical conditions, and are not designed to be optimally sparse. They might use 200 logical conditions when the same accuracy could be obtained with five logical conditions. (C4.5 and CART, decision trees suffer from these problems, as well as a vast number of models from the associative classification literature). An issue with algorithms that do not aim for optimal (or near-optimal) solutions to optimization problems is that it becomes difficult to tell whether poor performance is due to the choice of algorithm or the combination of the choice of model class and constraints. (Did the algorithm perform poorly because it did not optimize its objective, or because we chose constraints that do not allow enough flexibility in the model to fit the data well?) The question of computing optimal logical models has existed since at least the mid-1990s.

We would like models that look like they are created by hand, but they need to be accurate, full-blown ML models. To this end, let us consider the following optimization problem, which asks us to find a model that minimizes a combination of the fraction of misclassified training points and the size of the model. Training observations are indexed from i = 1, …, n, and is a family of logical models such as decision trees. The optimization problem is

Here, the size of the model can be measured by the number of logical conditions in the model, such as the number of leaves in a decision tree. The parameter λ is the classification error one would sacrifice in order to have one fewer term in the model; if λ is 0.01, it means we would sacrifice 1% training accuracy in order to reduce the size of the model by one. Another way to say this is that the model would contain an additional term only if this additional term reduced the error by at least 1%.

The optimization problem in equation () is generally known to be computationally hard. Versions of this optimization problem are some of the fundamental problems of artificial intelligence. The challenge is whether we can solve (or approximately solve) problems like this in practical ways, by leveraging new theoretical techniques and advances in hardware.

The model in Table is a ML model that comes from the CORELS algorithm. CORELS solves a special case of equation (), for the special choice of as the set of rule lists, and where the size of the model is measured by the number of rules in the list. Table has three ‘if–then’ rules so its size is 3. To minimize equation (), CORELS needs to avoid enumerating all possible models, because this would take an extremely long time (perhaps until the end of

the universe on a modern laptop for a fairly small data set). The technology underlying the CORELS algorithm was able to solve the optimization problem to optimality in under a minute for the Broward County, FL, data set discussed above. CORELS' backbone is (i) a set of theorems allowing massive reductions in the search space of rule lists, (ii) a custom fast bit-vector library that allows fast exploration of the search space, so that CORELS does not need to enumerate all rule lists and (iii) specialized data structures that keep track of intermediate computations and symmetries. This set of ingredients proved to be a powerful cocktail for handling these tough computational problems.

The example of CORELS enforces two points discussed above: (i) interpretable models sometimes entail hard computational problems and (ii) these computational problems can be solved by leveraging a combination of theoretical and systems-level techniques. CORELS creates one type of logical model, but there are many more. Formally, the first challenge is to create algorithms that solve logical modelling problems in a reasonable amount of time, for practical data sets.

We have been extending CORELS to more complex problems, such as falling rule lists, and optimal binary-split decision trees, but there is much work to be done on other types of logical model, with various kinds of constraints.

Note that it is possible to construct interpretable logical models for which the global model is large, and yet each explanation is small. This is discussed in Supplementary Section .

Challenge 2: construct optimal sparse scoring systems

Scoring systems have been designed by hand since at least the Burgess criminological model of 1928. The Burgess model was designed to predict whether a criminal would violate bail, where individuals received points for being a 'ne'er do well' or a 'recent immigrant' that increased their predicted probability of parole violation. (Of course, this model was not created using ML, which had not been invented yet.) A scoring system is a sparse linear model with integer coefficients—the coefficients are the point scores. An example of a scoring system for criminal recidivism is shown in Table , which predicts whether someone will be arrested within three years of release. Scoring systems are used pervasively throughout medicine; there are hundreds of scoring systems developed by physicians. Again, the challenge is whether scoring systems—which look like they could have been produced by a human in the absence of data— can be produced by a ML algorithm and be as accurate as any other model from any other ML algorithm.

Scoring system for risk of recidivism

1.	Prior arrests ? 2	1 point	?
2.	Prior arrests ? 5	1 point	+?
3.	Prior arrests for local ordinance	1 point	+?
4.	Age at release between 18 to 24	1 point	+?
5.	Age at release ? 40	?1 point	+?
		Score	= ?

Score	?1	0	1	2	3	4
Risk (%)	11.9	26.9	50.0	73.1	88.1	95.3

This system is from ref. , which was developed from refs. ,. The model was not created by a human; the selection of numbers and features come from the RiskSLIM machine learning algorithm.

There are several ways to formulate the problem of producing a scoring system (see, for example, refs. ,). For example, we could use a special case of equation (), where the model size is the number of terms in the model. (Table is a ML model with five terms.) Sometimes, one can round the coefficients of a logistic regression model to

produce a scoring system, but that method does not tend to give accurate models, and does not tend to produce models that have particularly nice coefficients (such as 1 and −1, used in Table). However, solving equation () or its variants is computationally hard, because the domain over which we solve the optimization problem is the integer lattice.

The model in Table arose from the solution to a very hard optimization problem, which is a variation of equation (). Let us discuss this optimization problem briefly. The goal is to find the coefficients bj, j = 1…p for the linear predictive model where zj is the jth covariate of a test observation z. In Table , the bj terms are the point scores, which turned out to be 1, −1 and 0 as a result of optimization, where only the non-zero coefficients are displayed in the figure. In particular, we want to solvewhere the point scores bj are constrained to be integers between −10 and 10, the training observations are indexed by i = 1, …, n, and p is the total number of covariates for our data. Here the model size is the number of non-zero coefficients, and again λ is the trade-off parameter. The first term is the logistic loss used in logistic regression. The problem is hard, specifically it is a mixed-integer-nonlinear program whose domain is the integer lattice.

Despite the hardness of this problem, new cutting plane algorithms have been able to solve this problem to optimality (or near-optimality) for arbitrarily large sample sizes and a moderate number of variables within a few minutes. The latest attempt at solving this problem is the RiskSLIM (Risk-Supersparse-Linear-Integer-Models) algorithm, which is a specialized cutting plane method that adds cutting planes only whenever the solution to a linear program is integer-valued, and otherwise performs branching.

This optimization problem is similar to what physicians attempt to solve manually, but without writing the optimization problem down like we did above. Because physicians do not use optimization tools to do this, accurate scoring systems tend to be difficult for physicians to create themselves from data. One of our collaborators spent months trying to construct a scoring system himself by adding and removing variables, rounding, and using other heuristics to decide which variables to add, remove and round. RiskSLIM was useful for helping him with this task. Formally, the second challenge is to create algorithms for scoring systems that are computationally efficient. Ideally we would increase the size of the optimal scoring system problems that current methods can practically handle by an order of magnitude.

Challenge 3: define interpretability for specific domains and create methods accordingly, including computer vision

Because interpretability needs to be defined in a domain-specific way, some of the most important technical challenges for the future are tied to specific important domains. Let us start with computer vision, for the classification of images. There is a vast and growing body of research on post hoc explainability of deep neural networks, but not as much work in designing interpretable neural networks. My goal in this section is to demonstrate that even for classic domains of ML, where latent representations of data need to be constructed, there could exist interpretable models that are as accurate as black box models.

For computer vision in particular, there is not a clear definition of interpretability, and the sparsity-related models discussed above do not apply—sparsity in pixel space does not make sense. There can be many different ideas of what constitutes interpretability, even between different computer vision applications. However, if we can define interpretability somehow for our particular application, we can embed this definition into our algorithm.

Let us define what constitutes interpretability by considering how people explain to each other the reasoning processes behind complicated visual classification tasks. As it turns out, for classification of natural images, domain experts often direct our attention to different parts of the image and explain why these parts of the image were important in their reasoning process.

The question is whether we can construct network architectures for deep learning that can also do this. The network must then make decisions by reasoning about parts of the image so that the explanations are real, and not post hoc.

In a recent attempt to do this, architectures that append a special prototype layer to the end of the network were built,. During training, the prototype layer finds parts of training images that act as prototypes for each class. For example, for bird classification, the prototype layer might pick out a prototypical head of a blue jay, prototypical feathers of a blue jay and so on. The network also learns a similarity metric between parts of images. Thus, during testing, when a new test image needs to be evaluated, the network finds parts of the test image that are similar to the prototypes it learned during training, as shown in Fig. . The final class prediction of the network is based on the weighted sum of similarities to the prototypes; this is the sum of evidence throughout the image for a particular class. The explanations given by the network are the prototypes (and the weighted similarities to them). These explanations are the actual computations of the model, and these are not post hoc explanations. The network is called ‘This look like that’ because its reasoning process considers whether ‘this’ part of the image looks like ‘that’ prototype.

Image from the authors of ref. , indicating that parts of the test image on the left are similar to prototypical parts of training examples.

The test image to be classified is on the left, the most similar prototypes are in the middle column, and the heatmaps that show which part of the test image is similar to the prototype are on the right. We included copies of the test image on the right so that it is easier to see to what part of the bird the heatmaps are referring. The similarities of the prototypes to the test image are what determine the predicted class label of the image. Here, the image is predicted to be a clay-coloured sparrow. The top prototype seems to be comparing the bird’s head to a prototypical head of a clay-coloured sparrow, the second prototype considers the throat of the bird, the third looks at feathers, and the last seems to consider the abdomen and leg. Credit: Image constructed by Alina Barnett, Duke University

Training this prototype network is not as easy as training an ordinary neural network; the tricks that have been developed for regular deep learning have not yet been developed for the prototype network. However, so far, these prototype networks have been trained to be approximately as accurate as the original black box deep neural networks from which they were derived, before the prototype layer was added.

Discussion on interpretability for specific domains

Let us finish this short discussion on challenges to interpretability for specific domains by mentioning that there are vast numbers of papers that have imbued interpretability in their methodology. Interpretability is not mentioned in the titles of these papers, and often not in the body of the text. This is why it is almost impossible to create a review article on interpretability in ML or statistics without missing the overwhelming majority of it.

It is not clear why it makes sense to create review articles for interpretability and explainability. We do not normally have reviews of performance/accuracy measures, despite the fact that there are many—accuracy, area under the receiver operating characteristic curve, partial area under the curve, sensitivity, specificity, discounted cumulative gain, F-score, G-means and many other domain-specific measures. Interpretability/explainability is just as domain-specific as accuracy performance, so it is not clear why reviews of interpretability make any more sense than reviews of accuracy/performance. I have yet to find even a single recent review that recognized the chasm between interpretability and explainability.

Let us discuss very briefly some examples of work on interpretability that would not have been covered by recent review articles, and yet are valuable contributions to interpretability in their respective domains. Gallagher et al. analyses brain-wide electrical spatiotemporal dynamics to understand depression vulnerability and find interpretable patterns in a low-dimensional space. Dimension reduction to interpretable dimensions is an important theme in interpretable ML. Problems residing in applied statistics are often interpretable because they embed the physics of the domain; for example, Wang et al. create models for recovery curves for prostatectomy patients whose signal and uncertainty obey specific constraints in order to be realistic. Constraints on the uncertainty of the predictions make these models interpretable.

The set-up of the recent 2018 FICO Explainable ML Challenge exemplified the blind belief in the myth of the accuracy/interpretability trade-off for a specific domain, namely credit scoring. Entrants were instructed to create a black box to predict credit default and explain the model afterwards. However, there was no performance difference between interpretable models and explainable models for the FICO data. A globally interpretable model won the FICO Recognition Prize for the competition. This is a case where the organizers and judges had not expected an interpretable model to be able to be constructed and thus did not ask entrants to try to construct such a model. The model of ref. was an additive model, which is a known form of interpretable model (see also refs. ,, where additive models are used for medical data). Additive models could be optimized using similar techniques to those introduced in Challenge 2 above.

A technical reason why accurate interpretable models might exist in many domains

Why is it that accurate interpretable models could possibly exist in so many different domains? Is it really possible that many aspects of nature have simple truths that are waiting to be discovered by ML? Although that would be intriguing, I will not make this kind of Occam's-razor-style argument, in favour of a technical argument about function classes, and in particular, Rashomon sets. The argument below will be fleshed out more formally in a manuscript my colleagues and I have in preparation. This is related to (but different from) the notation of 'flat minima', for which a nice example is given in ref. .

Here is the Rashomon set argument: consider that the data permit a large set of reasonably accurate predictive models to exist. Because this set of accurate models is large, it often contains at least one model that is interpretable. This model is thus both interpretable and accurate.

Unpacking this argument slightly, for a given data set, we define the Rashomon set as the set of reasonably accurate predictive models (say within a given accuracy from the best model accuracy of boosted decision trees). Because the data are finite, the data could admit many close-to-optimal models that predict differently from each other: a large Rashomon set. I suspect this happens often in practice because sometimes many different ML algorithms perform similarly on the same data set, despite having different functional forms (for example, random forests, neural networks, support vector machines). As long as the Rashomon set contains a large enough set of models with diverse predictions, it probably contains functions that can be approximated well by simpler functions, and so the Rashomon set can also contain these simpler functions. Said another way, uncertainty arising from the data leads to a Rashomon set; a larger Rashomon set probably contains interpretable models, thus interpretable accurate models often exist.

If this theory holds, we should expect to see interpretable models exist across domains. These interpretable models may be hard to find through optimization, but at least there is a reason we might expect that such models exist.

If there are many diverse yet good models, it means that algorithms may not be stable; an algorithm might choose one model, and a small change to that algorithm or to the data set may yield a completely different (but still accurate) model. This is not necessarily a bad thing, in fact, the availability of diverse good models means that domain experts may have more flexibility in choosing a model that they find interpretable. Supplementary Section discusses this in slightly more detail.

Conclusion

If this commentary can shift the focus even slightly from the basic assumption underlying most work in explainable ML—which is that a black box is necessary for accurate predictions—we will have considered this document a success.

If this document can encourage policy makers not to accept black box models without significant attempts at interpretable (rather than explainable) models, that would be even better.

If we can make people aware of the current challenges right now in interpretable ML, it will allow policy makers the mechanism to demand that more effort should be made in ensuring safety and trust in our ML models for high-stakes decisions.

If we do not succeed at these efforts, it is possible that black box models will continue to be permitted when it is not safe to use them. Because the definition of what constitutes a viable explanation is unclear, even strong regulations such as ‘right to explanation’ can be undermined with less-than-satisfactory explanations. Furthermore, there will continue to be problems combining black box model predictions with information outside the database, and continued miscalculations of black box model inputs. This may continue to lead to poor decisions throughout our criminal justice system, incorrect safety guidance for air quality disasters, incomprehensible loan decisions and other widespread societal problems.

Acknowledgements

The author thanks F. Wang, T. Wang, C. Chen, O. Li, A. Barnett, T. Dietterich, M. Seltzer, E. Angelino, N. Larus-Stone, E. Mannshart, M. Gupta and several others who helped my thought processes in various ways, and particularly B. Ustun, R. Parr, R. Holte and my father, S. Rudin, who went to considerable efforts to provide thoughtful comments and discussion. The author acknowledges funding from the Laura and John Arnold Foundation, NIH, NSF, DARPA, the Lord Foundation of North Carolina and MIT-Lincoln Laboratory.

Notes

Supplementary informationSupplementary information is available for this paper at *https://doi.org/10.1038/s42256-019-0048-x.Publisher’s* note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Load-Date: September 6, 2023

End of Document