A Roadmap for a Rigorous Science of Interpretability

Finale Doshi-Velez* and Been Kim*

From autonomous cars and adaptive email-filters to predictive policing systems, machine learning (ML) systems are increasingly ubiquitous; they outperform humans on specific tasks [Mnih et al., 2013, Silver et al., 2016, Hamill, 2017] and often guide processes of human understanding and decisions [Carton et al., 2016, Doshi-Velez et al., 2014]. The deployment of ML systems in complex applications has led to a surge of interest in systems optimized not only for expected task performance but also other important criteria such as safety [Otte, 2013, Amodei et al., 2016, Varshney and Alemzadeh, 2016], nondiscrimination [Bostrom and Yudkowsky, 2014, Ruggieri et al., 2010, Hardt et al., 2016], avoiding technical debt [Sculley et al., 2015], or providing the right to explanation [Goodman and Flaxman, 2016]. For ML systems to be used safely, satisfying these auxiliary criteria is critical. However, unlike measures of performance such as accuracy, these criteria often cannot be completely quantified. For example, we might not be able to enumerate all unit tests required for the safe operation of a semi-autonomous car or all confounds that might cause a credit scoring system to be discriminatory. In such cases, a popular fallback is the criterion of *interpretability*: if the system can *explain* its reasoning, we then can verify whether that reasoning is sound with respect to these auxiliary criteria.

Unfortunately, there is little consensus on what interpretability in machine learning *is* and how to *evaluate* it for benchmarking. Current interpretability evaluation typically falls into two categories. The first evaluates interpretability in the context of an application: if the system is useful in either a practical application or a simplified version of it, then it must be somehow interpretable (e.g. Ribeiro et al. [2016], Lei et al. [2016], Kim et al. [2015a], Doshi-Velez et al. [2015], Kim et al. [2015b]). The second evaluates interpretability via a quantifiable proxy: a researcher might first claim that some model class—e.g. sparse linear models, rule lists, gradient boosted trees—are interpretable and then present algorithms to optimize within that class (e.g. Bucilu et al. [2006], Wang et al. [2017], Wang and Rudin [2015], Lou et al. [2012]).

To large extent, both evaluation approaches rely on some notion of "you'll know it when you see it." Should we be concerned about a lack of rigor? Yes and no: the notions of interpretability above appear reasonable because they *are* reasonable: they meet the first test of having face-validity on the correct test set of subjects: human beings. However, this basic notion leaves many kinds of questions unanswerable: Are all models in all defined-to-be-interpretable model classes equally interpretable? Quantifiable proxies such as sparsity may seem to allow for comparison, but how does one think about comparing a model sparse in features to a model sparse in prototypes? Moreover, do all applications have the same interpretability needs? If we are to move this field forward—to compare methods and understand when methods may generalize—we need to formalize these notions and make them evidence-based.

The objective of this review is to chart a path toward the definition and rigorous evaluation of interpretability. The need is urgent: recent European Union regulation will *require* algorithms

*Authors contributed equally.

that make decisions based on user-level predictors, which "significantly affect" users to provide explanation ("right to explanation") by 2018 [Parliament and of the European Union, 2016]. In addition, the volume of research on interpretability is rapidly growing.[1] In section 1, we discuss what interpretability is and contrast with other criteria such as reliability and fairness. In section 2, we consider scenarios in which interpretability is needed and why. In section 3, we propose a taxonomy for the evaluation of interpretability—application-grounded, human-grounded and functionally-grounded. We conclude with important open questions in section 4 and specific suggestions for researchers doing work in interpretability in section 5.

1 What is Interpretability?

Definition Interpret means *to explain or to present in understandable terms.*[2] In the context of ML systems, we define interpretability as the *ability to explain or to present in understandable terms to a human.* A formal definition of explanation remains elusive; in the field of psychology, Lombrozo [2006] states "explanations are... the currency in which we exchanged beliefs" and notes that questions such as what constitutes an explanation, what makes some explanations better than others, how explanations are generated and when explanations are sought are just beginning to be addressed. Researchers have classified explanations from being "deductive-nomological" in nature [Hempel and Oppenheim, 1948] (i.e. as logical proofs) to providing some sense of mechanism [Bechtel and Abrahamsen, 2005, Chater and Oaksford, 2006, Glennan, 2002]. Keil [2006] considered a broader definition: implicit explanatory understanding. In this work, we propose data-driven ways to derive operational definitions and evaluations of explanations, and thus, interpretability.

Interpretability is used to confirm other important desiderata of ML systems There exist many auxiliary criteria that one may wish to optimize. Notions of *fairness* or *unbiasedness* imply that protected groups (explicit or implicit) are not somehow discriminated against. *Privacy* means the method protects sensitive information in the data. Properties such as *reliability* and *robustness* ascertain whether algorithms reach certain levels of performance in the face of parameter or input variation. *Causality* implies that the predicted change in output due to a perturbation will occur in the real system. *Usable* methods provide information that assist users to accomplish a task—e.g. a knob to tweak image lighting—while *trusted* systems have the confidence of human users—e.g. aircraft collision avoidance systems. Some areas, such as the fairness [Hardt et al., 2016] and privacy [Toubiana et al., 2010, Dwork et al., 2012, Hardt and Talwar, 2010] the research communities have formalized their criteria, and these formalizations have allowed for a blossoming of rigorous research in these fields (without the need for interpretability). However, in many cases, formal definitions remain elusive. Following the psychology literature, where Keil et al. [2004] notes "explanations may highlight an incompleteness," we argue that interpretability can assist in qualitatively ascertaining whether other desiderata—such as fairness, privacy, reliability, robustness, causality, usability and trust—are met. For example, one can provide a feasible explanation that fails to correspond to a causal structure, exposing a potential concern.

[1] Google Scholar finds more than 20,000 publications related to interpretability in ML in the last five years.
[2] Merriam-Webster dictionary, accessed 2017-02-07

2 Why interpretability? Incompleteness

Not all ML systems require interpretability. Ad servers, postal code sorting, air craft collision avoidance systems—all compute their output without human intervention. Explanation is not necessary either because (1) there are no significant consequences for unacceptable results or (2) the problem is sufficiently well-studied and validated in real applications that we trust the system's decision, even if the system is not perfect.

So when is explanation necessary and appropriate? We argue that the need for interpretability stems from an *incompleteness* in the problem formalization, creating a fundamental barrier to optimization and evaluation. Note that incompleteness is distinct from uncertainty: the fused estimate of a missile location may be uncertain, but such uncertainty can be rigorously quantified and formally reasoned about. In machine learning terms, we distinguish between cases where unknowns result in quantified variance—e.g. trying to learn from small data set or with limited sensors—and incompleteness that produces some kind of unquantified bias—e.g. the effect of including domain knowledge in a model selection process. Below are some illustrative scenarios:

- Scientific Understanding: The human's goal is to gain knowledge. We do not have a complete way of stating what knowledge is; thus the best we can do is ask for explanations we can convert into knowledge.

- Safety: For complex tasks, the end-to-end system is almost never completely testable; one cannot create a complete list of scenarios in which the system may fail. Enumerating all possible outputs given all possible inputs be computationally or logistically infeasible, and we may be unable to flag all undesirable outputs.

- Ethics: The human may want to guard against certain kinds of discrimination, and their notion of fairness may be too abstract to be completely encoded into the system (e.g., one might desire a 'fair' classifier for loan approval). Even if we can encode protections for specific protected classes into the system, there might be biases that we did not consider a priori (e.g., one may not build gender-biased word embeddings on purpose, but it was a pattern in data that became apparent only after the fact).

- Mismatched objectives: The agent's algorithm may be optimizing an incomplete objective—that is, a proxy function for the ultimate goal. For example, a clinical system may be optimized for cholesterol control, without considering the likelihood of adherence; an automotive engineer may be interested in engine data not to make predictions about engine failures but to more broadly build a better car.

- Multi-objective trade-offs: Two well-defined desiderata in ML systems may compete with each other, such as privacy and prediction quality [Hardt et al., 2016] or privacy and non-discrimination [Strahilevitz, 2008]. Even if each objectives are fully-specified, the exact dynamics of the trade-off may not be fully known, and the decision may have to be case-by-case.

In the presence of an incompleteness, explanations are one of ways to ensure that effects of gaps in problem formalization are visible to us.

3 How? A Taxonomy of Interpretability Evaluation

Even in standard ML settings, there exists a taxonomy of evaluation that is considered appropriate. In particular, the evaluation should match the claimed contribution. Evaluation of applied work should demonstrate success in the application: a game-playing agent might best a human player, a classifier may correctly identify star types relevant to astronomers. In contrast, core methods work should demonstrate generalizability via careful evaluation on a variety of synthetic and standard benchmarks.

In this section we lay out an analogous taxonomy of evaluation approaches for interpretability: application-grounded, human-grounded, and functionally-grounded. These range from task-relevant to general, also acknowledge that while human evaluation is essential to assessing interpretability, human-subject evaluation is not an easy task. A human experiment needs to be well-designed to minimize confounding factors, consumed time, and other resources. We discuss the trade-offs between each type of evaluation and when each would be appropriate.

3.1 Application-grounded Evaluation: Real humans, real tasks

Application-grounded evaluation involves conducting human experiments within a real application. If the researcher has a concrete application in mind—such as working with doctors on diagnosing patients with a particular disease—the best way to show that the model works is to evaluate it with respect to the task: doctors performing diagnoses. This reasoning aligns with the methods of evaluation common in the human-computer interaction and visualization communities, where there exists a strong ethos around making sure that the system delivers on its intended task [Antunes et al., 2012, Lazar et al., 2010]. For example, a visualization for correcting segmentations from microscopy data would be evaluated via user studies on segmentation on the target image task [Suissa-Peleg et al., 2016]; a homework-hint system is evaluated on whether the student achieves better post-test performance [Williams et al., 2016].

Specifically, we evaluate the quality of an explanation in the context of its end-task, such as whether it results in better identification of errors, new facts, or less discrimination. Examples of experiments include:

- Domain expert experiment with the exact application task.
- Domain expert experiment with a simpler or partial task to shorten experiment time and increase the pool of potentially-willing subjects.

In both cases, an important baseline is how well *human-produced* explanations assist in other humans trying to complete the task. To make high impact in real world applications, it is essential that we as a community respect the time and effort involved to do such evaluations, and also demand high standards of experimental design when such evaluations are performed. As HCI community recognizes [Antunes et al., 2012], this is not an easy evaluation metric. Nonetheless, it directly tests the objective that the system is built for, and thus performance with respect to that objective gives strong evidence of success.

3.2 Human-grounded Metrics: Real humans, simplified tasks

Human-grounded evaluation is about conducting simpler human-subject experiments that maintain the essence of the target application. Such an evaluation is appealing when experiments with the

target community is challenging. These evaluations can be completed with lay humans, allowing for both a bigger subject pool and less expenses, since we do not have to compensate highly trained domain experts. Human-grounded evaluation is most appropriate when one wishes to test more general notions of the quality of an explanation. For example, to study what kinds of explanations are best understood under severe time constraints, one might create abstract tasks in which other factors—such as the overall task complexity—can be controlled [Kim et al., 2013, Lakkaraju et al., 2016]

The key question, of course, is how we can evaluate the quality of an explanation without a specific end-goal (such as identifying errors in a safety-oriented task or identifying relevant patterns in a science-oriented task). Ideally, our evaluation approach will depend only on the quality of the explanation, regardless of whether the explanation is the model itself or a post-hoc interpretation of a black-box model, and regardless of the correctness of the associated prediction. Examples of potential experiments include:

- Binary forced choice: humans are presented with pairs of explanations, and must choose the one that they find of higher quality (basic face-validity test made quantitative).
- Forward simulation/prediction: humans are presented with an explanation and an input, and must correctly simulate the model's output (regardless of the true output).
- Counterfactual simulation: humans are presented with an explanation, an input, and an output, and are asked what must be changed to change the method's prediction to a desired output (and related variants).

Here is a concrete example. The common intrusion-detection test [Chang et al., 2009] in topic models is a form of the forward simulation/prediction task: we ask the human to find the difference between the model's true output and some corrupted output as a way to determine whether the human has correctly understood what the model's true output is.

3.3 Functionally-grounded Evaluation: No humans, proxy tasks

Functionally-grounded evaluation requires no human experiments; instead, it uses some formal definition of interpretability as a proxy for explanation quality. Such experiments are appealing because even general human-subject experiments require time and costs both to perform and to get necessary approvals (e.g., IRBs), which may be beyond the resources of a machine learning researcher. Functionally-grounded evaluations are most appropriate once we have a class of models or regularizers that have already been validated, e.g. via human-grounded experiments. They may also be appropriate when a method is not yet mature or when human subject experiments are unethical.

The challenge, of course, is to determine what proxies to use. For example, decision trees have been considered interpretable in many situations [Freitas, 2014]. In section 4, we describe open problems in determining what proxies are reasonable. Once a proxy has been formalized, the challenge is squarely an optimization problem, as the model class or regularizer is likely to be discrete, non-convex and often non-differentiable. Examples of experiments include

- Show the improvement of prediction performance of a model that is already proven to be interpretable (assumes that someone has run human experiments to show that the model class is interpretable).

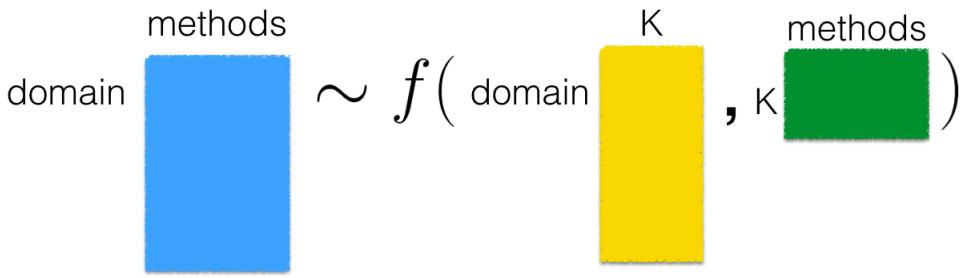

Figure 1: An example of data-driven approach to discover factors in interpretability

- Show that one's method performs better with respect to certain regularizers—for example, is more sparse—compared to other baselines (assumes someone has run human experiments to show that the regularizer is appropriate).

4 Open Problems in the Science of Interpretability, Theory and Practice

It is essential that the three types of evaluation in the previous section inform each other: the factors that capture the essential needs of real world tasks should inform what kinds of simplified tasks we perform, and the performance of our methods with respect to functional proxies should reflect their performance in real-world settings. In this section, we describe some important open problems for creating these links between the three types of evaluations:

1. What proxies are best for what real-world applications? (functionally to application-grounded)
2. What are the important factors to consider when designing simpler tasks that maintain the essence of the real end-task? (human to application-grounded)
3. What are the important factors to consider when characterizing proxies for explanation quality? (human to functionally-grounded)

Below, we describe a path to answering each of these questions.

4.1 Data-driven approach to discover factors of interpretability

Imagine a matrix where rows are specific real-world tasks, columns are specific methods, and the entries are the performance of the method on the end-task. For example, one could represent how well a decision tree of depth less than 4 worked in assisting doctors in identifying pneumonia patients under age 30 in US. Once constructed, methods in machine learning could be used to identify latent dimensions that represent factors that are important to interpretability. This approach is similar to efforts to characterize classification [Ho and Basu, 2002] and clustering problems [Garg and Kalai, 2016]. For example, one might perform matrix factorization to embed both tasks and methods respectively in low-dimensional spaces (which we can then seek to interpret), as shown in Figure 1. These embeddings could help predict what methods would be most promising for a new problem, similarly to collaborative filtering.

The challenge, of course, is in creating this matrix. For example, one could imagine creating a repository of clinical cases in which the ML system has access to the patient's record but not certain current features that are only accessible to the clinician, or a repository of discrimination-in-loan cases where the ML system must provide outputs that assist a lawyer in their decision. Ideally these would be linked to domain experts who have agreed to be employed to evaluate methods when applied to their domain of expertise. *Just as there are now large open repositories for problems in classification, regression, and reinforcement learning [Blake and Merz, 1998, Brockman et al., 2016, Vanschoren et al., 2014], we advocate for the creation of repositories that contain problems corresponding to real-world tasks in which human-input is required.* Creating such repositories will be more challenging than creating collections of standard machine learning datasets because they must include a system for human assessment, but with the availablity of crowdsourcing tools these technical challenges can be surmounted.

In practice, constructing such a matrix will be expensive since each cell must be evaluated in the context of a real application, and interpreting the latent dimensions will be an iterative effort of hypothesizing why certain tasks or methods share dimensions and then checking whether our hypotheses are true. In the next two open problems, we lay out some hypotheses about what latent dimensions may correspond to; these hypotheses can be tested via much less expensive human-grounded evaluations on simulated tasks.

4.2 Hypothesis: task-related latent dimensions of interpretability

Disparate-seeming applications may share common categories: an application involving preventing medical error at the bedside and an application involving support for identifying inappropriate language on social media might be similar in that they involve making a decision about a specific case—a patient, a post—in a relatively short period of time. However, when it comes to time constraints, the needs in those scenarios might be different from an application involving the understanding of the main characteristics of a large omics data set, where the goal—science—is much more abstract and the scientist may have hours or days to inspect the model outputs.

Below, we list a (non-exhaustive!) set of hypotheses about what might make tasks similar in their explanation needs:

- *Global vs. Local.* Global interpretability implies knowing what patterns are present in general (such as key features governing galaxy formation), while local interpretability implies knowing the reasons for a specific decision (such as why a particular loan application was rejected). The former may be important for when scientific understanding or bias detection is the goal; the latter when one needs a justification for a specific decision.

- *Area, Severity of Incompleteness.* What part of the problem formulation is incomplete, and how incomplete is it? We hypothesize that the types of explanations needed may vary depending on whether the source of concern is due to incompletely specified inputs, constraints, domains, internal model structure, costs, or even in the need to understand the training algorithm. The severity of the incompleteness may also affect explanation needs. For example, one can imagine a spectrum of questions about the safety of self-driving cars. On one end, one may have general curiosity about how autonomous cars make decisions. At the other, one may wish to check a specific list of scenarios (e.g., sets of sensor inputs that causes the car to drive off of the road by 10cm). In between, one might want to check a general property—safe urban driving—without an exhaustive list of scenarios and safety criteria.

- *Time Constraints.* How long can the user afford to spend to understand the explanation? A decision that needs to be made at the bedside or during the operation of a plant must be understood quickly, while in scientific or anti-discrimination applications, the end-user may be willing to spend hours trying to fully understand an explanation.

- *Nature of User Expertise.* How experienced is the user in the task? The user's experience will affect what kind of *cognitive chunks* they have, that is, how they organize individual elements of information into collections [Neath and Surprenant, 2003]. For example, a clinician may have a notion that autism and ADHD are both developmental diseases. The nature of the user's expertise will also influence what level of sophistication they expect in their explanations. For example, domain experts may expect or prefer a somewhat larger and sophisticated model—which confirms facts they know—over a smaller, more opaque one. These preferences may be quite different from hospital ethicist who may be more narrowly concerned about whether decisions are being made in an ethical manner. More broadly, decison-makers, scientists, compliance and safety engineers, data scientists, and machine learning researchers all come with different background knowledge and communication styles.

Each of these factors can be isolated in human-grounded experiments in simulated tasks to determine which methods work best when they are present.

4.3 Hypothesis: method-related latent dimensions of interpretability

Just as disparate applications may share common categories, disparate methods may share common qualities that correlate to their utility as explanation. As before, we provide a (non-exhaustive!) set of factors that may correspond to different explanation needs: Here, we define *cognitive chunks* to be the basic units of explanation.

- *Form of cognitive chunks.* What are the basic units of the explanation? Are they raw features? Derived features that have some semantic meaning to the expert (e.g. "neurological disorder" for a collection of diseases or "chair" for a collection of pixels)? Prototypes?

- *Number of cognitive chunks.* How many cognitive chunks does the explanation contain? How does the quantity interact with the type: for example, a prototype can contain a lot more information than a feature; can we handle them in similar quantities?

- *Level of compositionality.* Are the cognitive chunks organized in a structured way? Rules, hierarchies, and other abstractions can limit what a human needs to process at one time. For example, part of an explanation may involve *defining* a new unit (a chunk) that is a function of raw units, and then providing an explanation in terms of that new unit.

- *Monotonicity and other interactions between cognitive chunks.* Does it matter if the cognitive chunks are combined in linear or nonlinear ways? In monotone ways [Gupta et al., 2016]? Are some functions more natural to humans than others [Wilson et al., 2015, Schulz et al., 2016]?

- *Uncertainty and stochasticity.* How well do people understand uncertainty measures? To what extent is stochasticity understood by humans?

5 Conclusion: Recommendations for Researchers

In this work, we have laid the groundwork for a process to rigorously define and evaluate interpretability. There are many open questions in creating the formal links between applications, the science of human understanding, and more traditional machine learning regularizers. In the mean time, we encourage the community to consider some general principles.

The claim of the research should match the type of the evaluation. Just as one would be critical of a reliability-oriented paper that only cites accuracy statistics, the choice of evaluation should match the specificity of the claim being made. A contribution that is focused on a particular application should be expected to be evaluated in the context of that application (application-grounded evaluation), or on a human experiment with a closely-related task (human-grounded evaluation). A contribution that is focused on better optimizing a model class for some definition of interpretability should be expected to be evaluated with functionally-grounded metrics. As a community, we must be careful in the work on interpretability, both recognizing the need for and the costs of human-subject experiments.

We should categorize our applications and methods with a common taxonomy. In section 4, we hypothesized factors that may be the latent dimensions of interpretability. Creating a shared language around such factors is essential not only to evaluation, but also for the citation and comparison of related work. For example, work on creating a safe healthcare agent might be framed as focused on the need for explanation due to unknown inputs at the local scale, evaluated at the level of an application. In contrast, work on learning sparse linear models might also be framed as focused on the need for explanation due to unknown inputs, but this time evaluated at global scale. As we share each of our work with the community, we can do each other a service by describing factors such as

1. How is the problem formulation incomplete? (Section 2)
2. At what level is the evaluation being performed? (application, general user study, proxy; Section 3)
3. What are task-related relevant factors? (e.g. global vs. local, severity of incompleteness, level of user expertise, time constraints; Section 4.2)
4. What are method-related relevant factors being explored? (e.g. form of cognitive chunks, number of cognitive chunks, compositionality, monotonicity, uncertainty; Section 4.3)

and of course, adding and refining these factors as our taxonomies evolve. These considerations should move us away from vague claims about the interpretability of a particular model and toward classifying applications by a common set of terms.

Acknowledgments This piece would not have been possible without the dozens of deep conversations about interpretability with machine learning researchers and domain experts. Our friends and colleagues, we appreciate your support. We want to particularity thank Ian Goodfellow, Kush Varshney, Hanna Wallach, Solon Barocas, Stefan Rping and Jesse Johnson for their feedback.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Pedro Antunes, Valeria Herskovic, Sergio F Ochoa, and Jose A Pino. Structuring dimensions for collaborative systems evaluation. *ACM Computing Surveys*, 2012.

William Bechtel and Adele Abrahamsen. Explanation: A mechanist alternative. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*, 2005.

Catherine Blake and Christopher J Merz. {UCI} repository of machine learning databases. 1998.

Nick Bostrom and Eliezer Yudkowsky. The ethics of artificial intelligence. *The Cambridge Handbook of Artificial Intelligence*, 2014.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Cristian Bucilu, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006.

Samuel Carton, Jennifer Helsby, Kenneth Joseph, Ayesha Mahmud, Youngsoo Park, Joe Walsh, Crystal Cody, CPT Estella Patterson, Lauren Haynes, and Rayid Ghani. Identifying police officers at risk of adverse events. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.

Jonathan Chang, Jordan L Boyd-Graber, Sean Gerrish, Chong Wang, and David M Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, 2009.

Nick Chater and Mike Oaksford. Speculations on human causal learning and reasoning. *Information sampling and adaptive cognition*, 2006.

Finale Doshi-Velez, Yaorong Ge, and Isaac Kohane. Comorbidity clusters in autism spectrum disorders: an electronic health record time-series analysis. *Pediatrics*, 133(1):e54–e63, 2014.

Finale Doshi-Velez, Byron Wallace, and Ryan Adams. Graph-sparse lda: a topic model with structured sparsity. *Association for the Advancement of Artificial Intelligence*, 2015.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science Conference*. ACM, 2012.

Alex Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD Explorations*, 2014.

Vikas K Garg and Adam Tauman Kalai. Meta-unsupervised-learning: A supervised approach to unsupervised learning. *arXiv preprint arXiv:1612.09030*, 2016.

Stuart Glennan. Rethinking mechanistic explanation. *Philosophy of science*, 2002.

Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a" right to explanation". *arXiv preprint arXiv:1606.08813*, 2016.

Maya Gupta, Andrew Cotter, Jan Pfeifer, Konstantin Voevodski, Kevin Canini, Alexander Mangylov, Wojciech Moczydlowski, and Alexander Van Esbroeck. Monotonic calibrated interpolated look-up tables. *Journal of Machine Learning Research*, 2016.

Sean Hamill. CMU computer won poker battle over humans by statistically significant margin. `http://www.post-gazette.com/business/tech-news/2017/01/31/CMU-computer-won-poker-battle-over-humans-by-statistically-significant-margin/stories/201701310250`, 2017. Accessed: 2017-02-07.

Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *ACM Symposium on Theory of Computing*. ACM, 2010.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, 2016.

Carl Hempel and Paul Oppenheim. Studies in the logic of explanation. *Philosophy of science*, 1948.

Tin Kam Ho and Mitra Basu. Complexity measures of supervised classification problems. *IEEE transactions on pattern analysis and machine intelligence*, 2002.

Frank Keil. Explanation and understanding. *Annu. Rev. Psychol.*, 2006.

Frank Keil, Leonid Rozenblit, and Candice Mills. What lies beneath? understanding the limits of understanding. *Thinking and seeing: Visual metacognition in adults and children*, 2004.

Been Kim, Caleb Chacha, and Julie Shah. Inferring robot task plans from human team meetings: A generative modeling approach with logic-based prior. *Association for the Advancement of Artificial Intelligence*, 2013.

Been Kim, Elena Glassman, Brittney Johnson, and Julie Shah. iBCM: Interactive bayesian case model empowering humans via intuitive interaction. 2015a.

Been Kim, Julie Shah, and Finale Doshi-Velez. Mind the gap: A generative approach to interpretable feature selection and extraction. In *Advances in Neural Information Processing Systems*, 2015b.

Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1675–1684. ACM, 2016.

Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. *Research methods in human-computer interaction*. John Wiley & Sons, 2010.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*, 2016.

Tania Lombrozo. The structure and function of explanations. *Trends in cognitive sciences*, 10(10): 464–470, 2006.

Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Ian Neath and Aimee Surprenant. *Human Memory*. 2003.

Clemens Otte. Safe and interpretable machine learning: A methodological review. In *Computational Intelligence in Intelligent Data Analysis*. Springer, 2013.

Parliament and Council of the European Union. General data protection regulation. 2016.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. *arXiv preprint arXiv:1602.04938*, 2016.

Salvatore Ruggieri, Dino Pedreschi, and Franco Turini. Data mining for discrimination discovery. *ACM Transactions on Knowledge Discovery from Data*, 2010.

Eric Schulz, Joshua Tenenbaum, David Duvenaud, Maarten Speekenbrink, and Samuel Gershman. Compositional inductive biases in function learning. *bioRxiv*, 2016.

D Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems*, 2015.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 2016.

Lior Jacob Strahilevitz. Privacy versus antidiscrimination. *University of Chicago Law School Working Paper*, 2008.

Adi Suissa-Peleg, Daniel Haehn, Seymour Knowles-Barley, Verena Kaynig, Thouis R Jones, Alyssa Wilson, Richard Schalek, Jeffery W Lichtman, and Hanspeter Pfister. Automatic neural reconstruction from petavoxel of electron microscopy data. *Microscopy and Microanalysis*, 2016.

Vincent Toubiana, Arvind Narayanan, Dan Boneh, Helen Nissenbaum, and Solon Barocas. Adnostic: Privacy preserving targeted advertising. 2010.

Joaquin Vanschoren, Jan N Van Rijn, Bernd Bischl, and Luis Torgo. Openml: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.

Kush Varshney and Homa Alemzadeh. On the safety of machine learning: Cyber-physical systems, decision sciences, and data products. *CoRR*, 2016.

Fulton Wang and Cynthia Rudin. Falling rule lists. In *AISTATS*, 2015.

Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. Bayesian rule sets for interpretable classification. In *International Conference on Data Mining*, 2017.

Joseph Jay Williams, Juho Kim, Anna Rafferty, Samuel Maldonado, Krzysztof Z Gajos, Walter S Lasecki, and Neil Heffernan. Axis: Generating explanations at scale with learnersourcing and machine learning. In *ACM Conference on Learning@ Scale*. ACM, 2016.

Andrew Wilson, Christoph Dann, Chris Lucas, and Eric Xing. The human kernel. In *Advances in Neural Information Processing Systems*, 2015.