Learning to Explain Machine Learning

Vedant Nanda*
University of Maryland
MPI-SWS
vedant@cs.umd.edu

Duncan C. McElfresh*
University of Maryland
dmcelfre@umd.edu

John P. Dickerson
University of Maryland
john@cs.umd.edu

*Equal Contribution

CHI'20,, April 25–30, 2020, Honolulu, HI, USA
ACM 978-1-4503-6819-3/20/04.
https://doi.org/10.1145/3334480.XXXXXXX

Abstract
Explainable AI (XAI) methods yield human-understandable, often post-hoc descriptions of a machine learning (ML) model's behavior. Evaluation metrics for XAI methods fall within readily-measurable dimensions such as fidelity of the explanation to the underlying ML model, various forms of human comprehensibility, computational overhead, and others. We argue that—given ML models' role as only one piece of larger, deployed sociotechnical systems—these metrics alone do not enable the selection of an appropriate XAI method, or methods, for a specific use case. Indeed, it is necessary to include additional *context*, related to the user of the system as well as the downstream impact of the ML model. Inspired by prior work in human-computer interaction and computational social choice, we propose a learning-based framework for the selection of XAI methods that are tailored to each user and context.

Introduction
As machine learning (ML) models are increasingly used to influence important decisions, it is becoming increasingly important to *understand* the output of these models. This is especially important when the ML model is a "black box," and when the user is not an ML expert.

Researchers have proposed a variety of methods to explain the output of ML models, and a variety of properties to

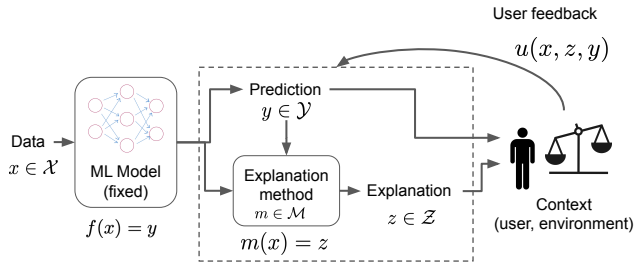

Figure 1: An overview of our framework. Data is passed to a fixed ML model, which outputs a prediction. The user receives the original data, the ML prediction, and the explanation, and assesses the usefulness in their decision-making scenario.

characterize their performance. Common properties of explanation methods include *fidelity* (how well the explanations match the underlying model), *stability* or *consistency* (whether similar inputs or similar models result in similar explanations), *comprehensibility* (whether the explanations are understood by an average person), and *computational complexity*, among others [6, 9, 22].

These properties help distinguish between different ML explanation methods, but they cannot indicate whether the resulting explanations are *useful* to the user. We believe that ML explanations should be tailored to *each* user, in *each* decision-making environment. We refer to the combination of user-and-environment as *context*: for example, an engineer debugging a marketing model might use complex explanations to understand aberrant behavior—this is one context. A policy analyst might interrogate the same marketing model to determine if it discriminates against certain customers—this is another context.

Context is defined variously in the HCI community but largely focuses on *information that can be used to characterize the situation of an entity [1, 5, 7, 8, 18]. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves*. Our definition fits this general definition of context.

Prior work has considered aspects of "good" ML explanations from a philosophical perspective [13], and through the lens of performance metrics [11]; another related concept is *personalized explanations* [19]. However, most prior work focuses on *fixed* explanation methods for a particular context; we consider an adaptive setting where the user or context may change.

We propose a framework for *learning* the most-useful explanations for each context through user feedback. First, we draw on prior work to emphasize why context matters.

Explanations based on Context

Context plays an important role in deployed ML systems: different users might prefer different explanations [26], or the same user might prefer different explanations for different use cases [11]. As examples, we discuss two well-studied phenomena—information overload and information presentation—that motivate user-specific preference modeling for explainability.

Information Overload [24] refers to the difficulty of effectively making decisions when presented with too much information. The amount of information presented by an explanation method can typically be controlled via the system designer. Take for example the class of feature attribution methods (e.g., SHAP [14], LIME [16], Saliency Maps [21], Guided Backprop [23]), which explain a given

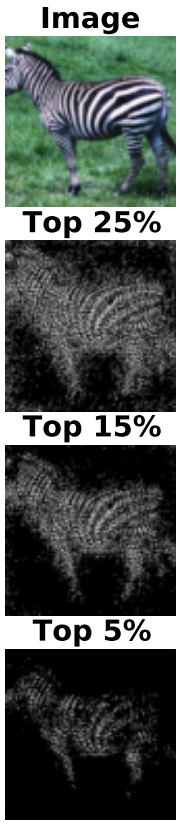

Figure 2: Explanation generated by guided backprop [23] for an image of a Zebra from ImageNet [17], for a ResNet50 model [10].

prediction by attributing weights to each input feature.[1] Highlighting *all* relevant features might overwhelm certain users, but highlighting only *some* features might not (for example, the $K\%$ most-important features). Figure 2 shows an example of how this could be done in a computer vision setting—and can lead to different looking explanations.

Information Presentation The format in which information is presented can have significant impacts on the end user [2, 20]. Different methods provide explanations in different formats. Feature-attribution-based methods highlight salient input features [14, 16, 21, 23] as shown in Figure 2; concept-based explanations [12] provide human-understandable concepts that were used to make predictions (e.g., a zebra was predicted because of stripes); and counterfactual explanations [25] provide guidelines on how the world must differ for some outcome to occur (e.g., an applicant would have been offered a loan if their income had been at least $50,000). Due to the inherent differences in the way information is presented, each class of methods can have varying amounts of usefulness for different use cases and different end-users.

These two examples illustrate that a useful explanation would depend on the *context*, *i.e.* both the user and the application for which an explanation is needed. In the next section, we provide a possible way to incorporate such information when explaining ML model output.

Mapping Explanations to Users

Prior work has identified several metrics for characterizing explanation methods and their output, including *fidelity*, *stability*, *comprehensibility*, and so on. The set of these metrics constitutes **embedding space** for the set of explanation methods and their outputs. To make context-aware explanations, we can learn which areas of this embedding are appropriate for different contexts by eliciting user feedback.

Consider for example the two-dimensional embedding in Figure 3: one axis represents *fidelity*, while the other represents *ease of communication* We hypothesize that different *regions* of this embedding space will be appropriate in different contexts; we propose learning which regions are useful in different contexts by eliciting user feedback. To make this idea more concrete, we formalize this process into a mathematical framework.

Framework

Our setting is summarized in Figure 1: an ML system and explanation method output both *predictions* and *explanations* to a user, who judges the usefulness of this output. To clarify, we use a small example.

Example. *A doctor is using a black-box ML system to diagnose a rare disease. The system takes as input a patient's electronic medical record (EMR) and outputs a probability of positive diagnosis. There are 10 different explanation methods available to explain the ML system output: each method returns the k features of the EMR (text fields) with the largest Shapley values, for $k = 1, \ldots, 10$.*

We use a mathematical framework to reason in this setting. Let $\mathcal{X}$ and $\mathcal{Y}$ be the sets of input and output labels of the ML system. In our example, $\mathcal{X}$ is a set of EMRs, and $\mathcal{Y} \equiv [0, 1]$ is the probability of a positive diagnosis. Let $f : \mathcal{X} \to \mathcal{Y}$ denote the prediction function of the ML system: $f(x)$ is the predicted label for data point x.

Let $\mathcal{M}$ denote the set of possible explanation methods, and

[1] In line with the message of this paper, we make no specific claims about the efficacy of specific explainability methods; rather, methods mentioned by name serve as examples of popular approaches to explainable ML—complete with their pros and cons, known and unknown.

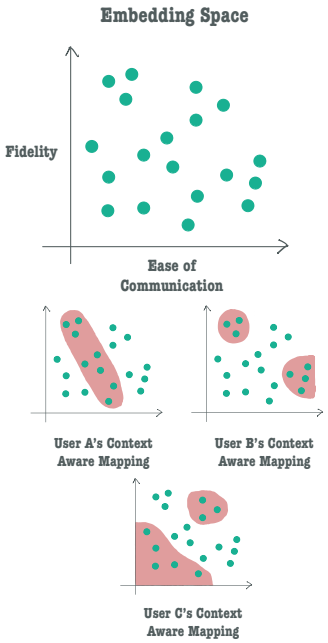

Figure 3: Hypothetical mapping of explanation methods to an embedding space. In this example embedding space is defined by *fidelity* and *ease of communication*. Each explanation method corresponds to a point in this space, and we aim to model different user's context-dependent preferences. The highlighted regions show a hypothetical mapping to each user's preference function over the embedding space. That is, in the given context, explanations from the highlighted region will result in maximum utility for the user.

let $\mathcal{Z}$ denote the set of possible *explanations* returned by these methods. In our example, $\mathcal{M} \equiv \{m_1, \ldots, m_{10}\}$, where m_k returns the k features with the largest Shapley values; in this example, the set of explanations $\mathcal{Z}$ is the set of all subsets of EMR features, up to size 10.

Each explanation method $m \in \mathcal{M}$ is itself a function $m : \mathcal{X} \to \mathcal{Z}$, where $m(x)$ is the explanation for data point x.[2] Using our example, $m_2(x)$ would return the set of two features with the greatest Shapley values for input x. Finally, the user considers the input data, the prediction, and the explanation, and determines the *usefulness* of this output.

Modeling Explanations We hypothesize that users will judge the usefulness of explanations along the many criteria (or metrics) developed by the HCI and ML communities. For this reason, we represent each explanation z as a point in the **embedding space** described in the previous section. For example, explanations returned by m_1 might have a high "ease of communication score" but a low "fidelity" score, while explanations from m_{10} might have lower "ease of communication" and greater "fidelity."

Modeling the User We represent the user's perception of usefulness with a *utility function* $u : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to [0, 1]$, which is a standard tool for modeling preferences. In a deployed setting we would use user feedback to learn a numerical representation of u; here we adapt techniques from *preference elicitation* (e.g., [3]). There are two standard approaches for learning u, which use different types of questions. We can learn *relative* utility by using *comparison questions*: we ask the user which input-output tuple is more useful, (x, y, z) or (x', y', z'). If (x, y, x) is more useful, then we learn the constraint $u(x, y, s) > u(x', y', z')$, otherwise $u(x, y, s) < u(x', y', z')$. Alternatively, we can learn *absolute* utility with Likert-style questions, such as *"on a scale from 1 to 5, how useful is* (x, y, z)*"*; responses to these questions map directly to numerical values, $u(x, y, z)$.

[2] Explanation methods usually depend on the ML model; in our setting the ML model is fixed, and this dependence is implicit.

Generating Context-Aware Explanations After learning a model of user utility in a decision-making context, $u(x, y, z)$, our framework can be used to generate context-aware explanations. When presented with a new data point x and ML output y, we identify the *most useful* explanation to the user by solving the problem $\max_{z \in \mathcal{Z}} u(x, y, z)$.

Discussion & Steps Forward

This position paper is conceptual in nature but outlines a framework that could be implemented directly using present-day elicitation and learning techniques from the computational social choice literature [4]. We note that, in many industry settings, our proposed framework forms a component of yet a larger system—a *deployment context*, that changes based on interaction with the outside world, shifting business constraints and goals, advances in technology, rotation of employees and clients, and so on.

In the face of a changing production environment, ML models get deployed and largely remain static; metrics degrade over time due to data drift, upstream and downstream changes, and human error. We would expect a user's context, and thus their internal and expressed preferences, to change over time as well based not just on those dynamics but also by virtue of discovering their preferences via interaction with the system itself [15]. Thus, one core direction for future research would be the incorporation of this larger "business lifecycle" into the context we address in our present framework.

Acknowledgments

The authors were supported in part by NSF CAREER Award IIS-1846237, NSF D-ISN Award #2039862, NSF Award CCF-1852352, NIH R01 Award NLM-013039-01, NIST MSE Award #20126334, DARPA GARD #HR00112020007, DoD WHS Award #HQ003420F0035, and a Google Faculty Research Award. Nanda was also supported by an ERC Advanced Grant “Foundations for Fair Social Computing” (no. 789373). The authors would like to thank Preethi Lahoti for useful comments on a draft of the paper and Krishna Gummadi for useful discussions.

REFERENCES

[1] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. 1999. Towards a Better Understanding of Context and Context-Awareness. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing (HUC '99)*. Springer-Verlag, Berlin, Heidelberg, 304–307.

[2] James R. Bettman and Pradeep Kakkar. 1977. Effects of Information Presentation Format on Consumer Information Acquisition Strategies. *Journal of Consumer Research* 3, 4 (03 1977), 233–240. DOI: http://dx.doi.org/10.1086/208672

[3] Avrim Blum, Jeffrey Jackson, Tuomas Sandholm, and Martin Zinkevich. 2004. Preference elicitation and query learning. *Journal of Machine Learning Research* 5, Jun (2004), 649–667.

[4] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. 2016. *Handbook of computational social choice*. Cambridge University Press.

[5] Peter J Brown, John D Bovey, and Xian Chen. 1997. Context-aware applications: from the laboratory to the marketplace. *IEEE personal communications* 4, 5 (1997), 58–64.

[6] Nadia Burkart and Marco F Huber. 2021. A Survey on the Explainability of Supervised Machine Learning. *Journal of Artificial Intelligence Research* 70 (2021), 245–317.

[7] Anind K Dey. 1998. Context-aware computing: The CyberDesk project. In *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*. 51–54.

[8] Anind K Dey. 2001. Understanding and using context. *Personal and ubiquitous computing* 5, 1 (2001), 4–7.

[9] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)* 51, 5 (2018), 1–42.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[11] Sérgio Jesus, Catarina Belém, Vladimir Balayan, João Bento, Pedro Saleiro, Pedro Bizarro, and João Gama. 2021. How can I choose an explainer? An Application-grounded Evaluation of Post-hoc Explanations. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*.

[12] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory sayres. 2018. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80. 2668–2677. http://proceedings.mlr.press/v80/kim18d.html

[13] Joy Lu, Dokyun (DK) Lee, Tae Wan Kim, and David Danks. 2020. Good Explanation for Algorithmic Transparency. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES '20)*. Association for Computing Machinery, New York, NY, USA, 93. DOI: http://dx.doi.org/10.1145/3375627.3375821

[14] Scott Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874* (2017).

[15] Charles Plott. 1993. *Rational Individual Behavior in Markets and Social Choice Processes*. Technical Report. California Institute of Technology, Division of the Humanities and Social ....

[16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[17] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. (2015).

[18] Nick S Ryan, Jason Pascoe, and David R Morse. 1998. Enhanced reality fieldwork: the context-aware archaeological assistant. In *Computer applications in archaeology*. Tempus Reparatum.

[19] Johanes Schneider and Joshua Handali. 2019. Personalized explanation in machine learning: A conceptualization. In *In Proceedings of the 27th European Conference on Information Systems (ECIS)*.

[20] Ben Shneiderman and Catherine Plaisant. 2010. *Designing the user interface: Strategies for effective human-computer interaction*. Pearson Education India.

[21] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013).

[22] Kacper Sokol and Peter Flach. 2020. Explainability fact sheets: a framework for systematic assessment of explainable approaches. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. 56–67.

[23] J.T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. 2015. Striving for Simplicity: The All Convolutional Net. In *ICLR (workshop track)*. http://lmb.informatik.uni-freiburg.de/Publications/2015/DB15a

[24] Alvin Toffler. 1970. Future shock, 1970. *Sydney. Pan* (1970).

[25] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.* 31 (2017), 841.

[26] Yishan Zhou and David Danks. 2020. Different "Intelligibility" for Different Folks. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES '20)*. New York, NY, USA, 194–199. DOI: http://dx.doi.org/10.1145/3375627.3375810