# Model-Agnostic Interpretability of Machine Learning

**Marco Tulio Ribeiro** MARCOTCR@CS.UW.EDU
**Sameer Singh** SAMEER@CS.UW.EDU
**Carlos Guestrin** GUESTRIN@CS.UW.EDU
University of Washington Seattle, WA 98195 USA

## Abstract

Understanding why machine learning models behave the way they do empowers both system designers and end-users in many ways: in model selection, feature engineering, in order to *trust* and act upon the predictions, and in more intuitive user interfaces. Thus, interpretability has become a vital concern in machine learning, and work in the area of interpretable models has found renewed interest. In some applications, such models are as accurate as non-interpretable ones, and thus are preferred for their transparency. Even when they are not accurate, they may still be preferred when interpretability is of paramount importance. However, restricting machine learning to interpretable models is often a severe limitation. In this paper we argue for explaining machine learning predictions using *model-agnostic* approaches. By treating the machine learning models as black-box functions, these approaches provide crucial flexibility in the choice of models, explanations, and representations, improving debugging, comparison, and interfaces for a variety of users and models. We also outline the main challenges for such methods, and review a recently-introduced model-agnostic explanation approach (LIME) that addresses these challenges.

## 1. Introduction

As machine learning becomes a crucial component of an ever-growing number of user-facing applications, *interpretable machine learning* has become an increasingly important area of research for a number of reasons. First, as humans are the ones who train, deploy, and often use the predictions of machine learning models in the real world, it is of utmost importance for them to be able to trust the model. Apart from indicators such as accuracy on sample instances, a user's trust is directly impacted by how much they can understand and predict the model's behavior, as opposed to treating it as a black box. Second, a system designer who understands why their model is making predictions is certainly better equipped to improve it by means of feature engineering, parameter tuning, or even by replacing the model with a different one. Lastly, even in lower stakes domains such as movie or book recommendations, getting a rationale such as "you will probably like this book because of your interest in Russian Literature" makes the model much more useful to the users, and more likely to be trusted. Thus there is a crucial need to be able to explain machine learning predictions, i.e. provide users a rationale for why a prediction was made using textual and visual components of the data, and/or producing counter-factual knowledge of what would happen were the components different.

The prevailing solution to this explanation problem is to use so called "interpretable" models, such as decision trees, rules (Letham et al., 2015; Wang & Rudin, 2015), additive models (Caruana et al., 2015), attention-based networks (Xu et al., 2015), or sparse linear models (Ustun & Rudin, 2015). Instead of supporting models that are functionally black-boxes, such as an arbitrary neural network or random forests with thousands of trees, these approaches use models in which there is the possibility of meaningfully inspecting model components directly — e.g. a path in a decision tree, a single rule, or the weight of a specific feature in a linear model. As long as the model is accurate for the task, and uses a reasonably restricted number of internal components (i.e. paths, rules, or features), such approaches provide extremely useful insights.

An alternative approach to interpretability in machine learning is to be *model-agnostic*, i.e. to extract post-hoc explanations by treating the original model as a black box. This involves learning an interpretable model on the predictions of the black box model (Craven & Shavlik, 1996; Baehrens et al., 2010), perturbing inputs and seeing how the black box model reacts (Strumbelj & Kononenko, 2010; Krause et al., 2016), or both (Ribeiro et al., 2016).

*2016 ICML Workshop on Human Interpretability in Machine Learning (WHI 2016)*, New York, NY, USA. 

In this position paper, we argue for separating explanations from the model (i.e. being model agnostic). The summary of our position is that restricting the space of models to be interpretable is a constraint that results in less flexibility, accuracy, and usability. We develop this position with examples, while also describing the inherent challenges in model agnosticism. Finally, we review the recently-introduced LIME approach (Ribeiro et al., 2016), and discuss how it provides many of the desirable characteristics for model-agnostic explanations.

## 2. A Case for Model Agnosticism

In this section, we make a case for model-agnostic interpretability, as opposed to just using interpretable models.

### 2.1. Model Flexibility

For most real-world applications, it is necessary to train models that are accurate for the task, irrespective of how complex or uninterpretable the underlying mechanism may be. We can observe this ideology manifesting with the increasing commonplace deployment of uninterpretable deep neural architectures for a wide variety of tasks.

Interpretable models for such tasks remain unsatisfying; such models are inherently crippled by the need to be understandable, being susceptible to the limited "perception budget" (Miller, 1956) of the users. This trade-off between model flexibility and interpretability (Freitas, 2014) implies one cannot use a model whose behavior is very complex, yet expect humans to fully comprehend it globally. For example, for a task such as predicting the sentiment of a sentence, producing an accurate model that is understandable seems like an unfeasible task. The size of the vocabulary alone makes it impossible for a short set of rules, a decision tree, or an additive model to be sufficiently accurate, not to mention more complex word interactions such as negation. Tasks that involve sensory data, such as audio and images, also suffer from the same problem: for a model to be useful, it must be sufficiently flexible to handle the data complexity.

In model-agnostic interpretability, the model is treated as a black box. The separation of interpretability from the model thus frees up the model to be as flexible as necessary for the task, enabling the use of any machine learning approach - including, for example, arbitrary deep neural networks. It also allows for the control of the complexity-interpretability trade-off (see next section), or "failing gracefully" if an interpretable explanation is not possible.

### 2.2. Explanation Flexibility

Different kinds of explanations meet different information needs. In some cases, users may only care about positive evidence towards a certain prediction (e.g. which part of an image is most responsible for the prediction), while in other instances knowing the negative evidence may be useful (e.g. in debugging a classifier). Yet in other cases, the information need may be of counter-factuals, e.g. how the model would behave if certain features had different values. Different users may also be able to handle different kinds of explanations; a user trained in statistics may be able to understand a Bayesian network, while a linear model is more intuitive to the layman. Even if the explanation type is kept fixed, users may tolerate different granularities in different situations. For example, Freitas (2014) notes a case where 41 rules are considered overwhelming, and contrasts it to another user who patiently analyzed 29,050 rules.

Most interpretable models are, however, restricted in what explanations are possible, be it a prototype (Kim et al., 2014), a set of rules (Letham et al., 2015) or line graphs (Caruana et al., 2015). Further, other constraints on interpretability, such as granularity, also have to be set *a priori* (e.g. max number of rules). On the other hand, by keeping the model separate from the explanations, one is able to tailor the explanation to the information need, while keeping the model fixed. If it is possible to measure how faithful the explanation is to the original model, one can effectively control the trade-off between fidelity and interpretability, as favored by Freitas (2014). Such approaches may also be able to provide multiple explanations of different types to the user, perhaps automatically picking the one with the highest faithfulness. Thus, by being model-agnostic, the same model can be explained with different types of explanations, and different degrees of interpretability for each type of explanation.

### 2.3. Representation Flexibility

In domains such as images, audio and text, many of the features used to represent instances in state-of-the-art solutions are themselves not interpretable. Unsupervised feature learning produces representations such as word embeddings (Mikolov et al., 2013), or the so-called deep features (Zhou et al., 2014). While an interpretable model trained on such features is still uninterpretable, model-agnostic approaches can generate explanations using different features than the one used by the underlying model. Thus, even if the model is using word embeddings, the explanations can be in terms of words, for example.

### 2.4. Lower Cost to Switch

Switching models is not an uncommon operation in machine learning pipelines. If one commits to using an interpretable model, one is "locked-in" to a particular model and a particular kind of explanations - even if newer, more accurate models are developed. Even when the switch is from one interpretable model to another, users may have to

be re-trained in understanding the new explanations, and the model's utility may decrease due to cognitive overhead. In contrast, if one uses model-agnostic explanations, switching the underlying model for a new one is trivial, while the way in which the explanations are presented is maintained.

### 2.5. Comparing Two Models

When deploying machine learning in the real world, a system designer often has to decide between one or more contenders, and an incumbent model. This comparison is hard to do if any of the systems are using interpretable models, while others are not. Further, even if all of the models are interpretable, it may still be difficult to compare the insights gained from each if the underlying explanations are different in their representation - for example comparing a rule-based model with a tree-based model. It is also not clear what to do if one of the contenders is less accurate but more interpretable, or vice versa. With model-agnostic explanations, the models being compared can be explained using the same techniques and representations.

## 3. Challenges for Model-agnostic Explanations

While we have made a case for model agnosticism, this approach is not without its challenges. For example, getting a global understanding of the model may be hard if the model is very complex, due to the trade-off between flexibility and interpretability. To make matters worse, local explanations may be inconsistent with one another, since a flexible model may use a certain feature in different ways depending on the other features. In Ribeiro et al. (2016) we explained text models by selecting a small number of representative and non-redundant individual prediction explanations obtained via submodular optimization, similar in spirit to showing prototypes (Kim et al., 2014). However, it is unclear on how to extend this approach to domains such as images or tabular data, where the data itself is not sparse.

In some domains, exact explanations may be required (e.g. for legal or ethical reasons), and using a black-box may be unacceptable (or even illegal). Interpretable models may also be more desirable when interpretability is much more important than accuracy, or when interpretable models trained on a small number of carefully engineered features are as accurate as black-box models.

Another challenge for model-agnostic explanations is to be actionable. Using a white box makes it easier to incorporate user feedback in systems like iBCM (Kim et al., 2015), or injecting logic into matrix factorization (Rocktaschel et al., 2015). Feature labeling (Druck et al., 2008) or annotator rationales (Zaidan & Eisner, 2008) are other forms of feedback that should be supported for explanations. A basic form of feature engineering (removing bad features) via explanations has been shown to be effective (Ribeiro et al., 2016), but incorporating more powerful forms of feedback from the users is still a challenging research direction, in particular while remaining model-agnostic.

## 4. Local Interpretable Model-agnostic Explanations (LIME)

We now briefly review LIME (Ribeiro et al., 2016), and discuss how it maintains model-agnosticism, while addressing some of the challenges that are described in the previous section. We denote $x \in \mathbb{R}^d$ as the original representation of an instance being explained, and we use $x' \in \mathbb{R}^{d'}$ to denote a vector for its interpretable representation. As exemplified before, $x$ may be a feature vector containing word embeddings, with $x'$ being the bag of words.

LIME's goal is to identify an **interpretable** model over the *interpretable representation* that is **locally faithful** to the classifier. Even though an interpretable model may not be able to approximate the black box model globally, approximating it in the vicinity of an individual instance may be feasible. Formally, the explanation model is $g : \mathbb{R}^{d'} \rightarrow \mathbb{R}, g \in G$, where $G$ is a class of potentially interpretable models, such as linear models, decision trees, or rule lists, i.e. given a model $g \in G$, we can present it to the user as an explanation with visual or textual artifacts. As noted before, not every $g \in G$ is simple enough to be interpretable - thus we let $\Omega(g)$ be a measure of complexity (as opposed to interpretability) of $g$, which may be either a soft constraint (e.g. the depth of a tree, or the number of non-zeros in a linear model) or a hard constraint (e.g. $\infty$ if the depth or the number of non-zeros is above a certain threshold).

Let the model being explained be $f : \mathbb{R}^d \rightarrow \mathbb{R}$, e.g. in classification $f(x)$ is the probability that $x$ belongs to a certain class. We further use $\Pi_x(z)$ as a proximity measure between an instance $z$ to $x$, so as to define locality around $x$. Finally, let $\mathcal{L}(f, g, \Pi_x)$ be a measure of how unfaithful $g$ is in approximating $f$ in the locality defined by $\Pi_x$. In order to ensure both interpretability and local fidelity, we must minimize $\mathcal{L}(f, g, \Pi_x)$ while having $\Omega(g)$ be low enough to be interpretable by humans. The explanation $\xi(x)$ produced by **LIME** is obtained by solving:

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \quad \mathcal{L}(f, g, \Pi_x) + \Omega(g) \tag{1}$$

This formulation can be used with different explanation families $G$, fidelity functions $\mathcal{L}$, and complexity measures $\Omega$. We estimate $\mathcal{L}$ by generating perturbed samples around $x$, making predictions with the black box model $f$ and weighting them according to $\Pi_x$. The intuition for this is presented in Figure 1, where a globally complex model is explained using a locally-faithful linear explanation.

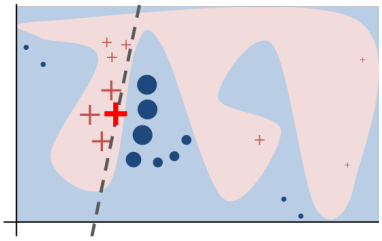
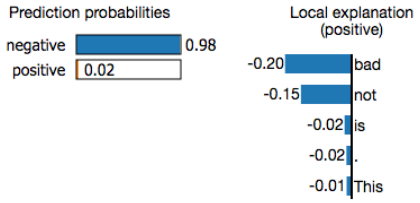


*Figure 1.* Toy example to present intuition for LIME. The black-box model's complex decision function $f$ (unknown to LIME) is represented by the blue/pink background. The bright bold red cross is the instance being explained. LIME samples instances, gets predictions using $f$, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the explanation that is locally (but not globally) faithful.




(a) Logistic Regression trained on unigrams


(b) LSTM trained on sentence embeddings.

*Figure 2.* Explaining sentiment predictions for the sentence "This is not bad.", using different models and representations

**Discussion**

Some approaches are model agnostic by approximating the black box model by an interpretable one globally (Craven & Shavlik, 1996; Baehrens et al., 2010; Sanchez et al., 2015). Global explanation, however, are often either not interpretable, or too simplistic to represent the original model. LIME's focus on explaining individual predictions allows more accurate explanations while retaining **model flexibility**. For example, it is easy to explain why sentences such as "This is not bad." have a positive sentiment, even if we are not able to explain the complete sentiment model.

For **explanation flexibility**, the practitioner has complete control over $G$ and $\Omega(g)$; in Ribeiro et al. (2016), for example, we use very sparse linear models. This representation is simple enough for non-expert Mechanical Turkers to perform model selection and feature engineering effectively for complex, uninterpretable models. Furthermore, since LIME estimates the local fidelity through $\mathcal{L}$, we can directly control the interpretability of the explanations (e.g. using as many words as needed to maintain faithfulness) or whether to only display interpretable explanations when they are accurate to the black box model. LIME also supports exploring multiple explanation families $G$ simultaneously, and picking the one with highest faithfulness.

**Representation flexibility** is built into LIME, with the distinction between original $x$ and interpretable representation $x'$. In Ribeiro et al. (2016), we explain models trained on on word embeddings by using words as interpretable representation, and a neural network trained on raw pixels by using contiguous super-pixels as $x'$.

We demonstrate the small **switching costs** of LIME by explaining a wide variety of models (random forests, SVMs, neural networks, linear models, and nearest neighbors) using the same type of explanations. We also demonstrate LIME's utility for model comparison by enabling non-expert Mechanical Turk users to select which of two competing models would generalize better using the explanations.

As a final illustration, we explain the predictions two sentiment analysis classifiers on the sentence "This is not bad.", using the class of linear models as $G$. The classifiers vary wildly in complexity and underlying representation - one is a logistic regression trained on unigrams, while the other an LSTM neural network trained on sentence embeddings (Wieting et al., 2015). Explanations, given in terms of words (and their associated weights in a bar chart) in Figure 2, demonstrate that completely different classifiers can be described in a unified, interpretable manner. In Figure 2(b), the explanation assigns positive weight to both "not" and "bad", as only the conjunction is responsible for the LSTM's positive prediction (even though interactions are not modeled explicitly).

## 5. Conclusion

Although interpretable models provide crucial insight into why predictions are made, they impose restrictions on the model, representation (features), and the expertise of the users. We argued that model-agnostic explanation systems provide a generic framework for interpretability that allows for flexibility in the choice of models, representations, and the user expertise. We outlined a number of challenges that need to be addressed for model-agnostic approaches; some of which are addressed by the recently introduced LIME (Ribeiro et al., 2016), while others are left as future work. We thus conclude that model-agnostic interpretability is a key component in making machine learning more trustworthy - and ultimately, more useful.

## Acknowledgements

This work was supported in part by ONR awards #W911NF-13-1-0246 and #N00014-13-1-0023, and in part by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

## References

Baehrens, David, Schroeter, Timon, Harmeling, Stefan, Kawanabe, Motoaki, Hansen, Katja, and Müller, Klaus-Robert. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11, 2010.

Caruana, Rich, Lou, Yin, Gehrke, Johannes, Koch, Paul, Sturm, Marc, and Elhadad, Noemie. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Knowledge Discovery and Data Mining (KDD)*, 2015.

Craven, Mark W and Shavlik, Jude W. Extracting tree-structured representations of trained networks. *Advances in neural information processing systems*, pp. 24–30, 1996.

Druck, Gregory, Mann, Gideon, and McCallum, Andrew. Learning from labeled features using generalized expectation criteria. In *ACM SIGIR conference on Research and development in information retrieval*, pp. 595–602. ACM, 2008.

Freitas, Alex A. Comprehensible classification models: A position paper. *SIGKDD Explor. Newsl.*, 15(1):1–10, March 2014. ISSN 1931-0145.

Kim, Been, Rudin, Cynthia, and Shah, Julie A. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 1952–1960. Curran Associates, Inc., 2014.

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

Krause, Josua, Perer, Adam, and Ng, Kenney. Interacting with predictions: Visual inspection of black-box machine learning models. 2016.

Letham, Benjamin, Rudin, Cynthia, McCormick, Tyler H., and Madigan, David. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *Annals of Applied Statistics*, 2015.

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg S, and Dean, Jeff. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems (NIPS)*. 2013.

Miller, George. The magical number seven, plus or minus two: Some limits on our capacity for processing information, 1956.

Ribeiro, Marco Tulio, Singh, Sameer, and Guestrin, Carlos. "why should I trust you?": Explaining the predictions of any classifier. In *Knowledge Discovery and Data Mining (KDD)*, 2016.

Rocktaschel, Tim, Singh, Sameer, and Riedel, Sebastian. Injecting logical background knowledge into embeddings for relation extraction. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2015.

Sanchez, Ivan, Rocktaschel, Tim, Riedel, Sebastian, and Singh, Sameer. Towards extracting faithful and descriptive representations of latent variable models. In *AAAI Spring Syposium on Knowledge Representation and Reasoning (KRR): Integrating Symbolic and Neural Approaches*, 2015.

Strumbelj, Erik and Kononenko, Igor. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11, 2010.

Ustun, Berk and Rudin, Cynthia. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 2015.

Wang, Fulton and Rudin, Cynthia. Falling rule lists. In *Artificial Intelligence and Statistics (AISTATS)*, 2015.

Wieting, John, Bansal, Mohit, Gimpel, Kevin, and Livescu, Karen. Towards universal paraphrastic sentence embeddings. *CoRR*, abs/1511.08198, 2015.

Xu, Kelvin, Ba, Jimmy, Kiros, Ryan, Cho, Kyunghyun, Courville, Aaron, Salakhutdinov, Ruslan, Zemel, Richard, and Bengio, Yoshua. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning (ICML)*, 2015.

Zaidan, Omar F. and Eisner, Jason. Modeling annotators: A generative approach to learning from annotator rationales. In *EMNLP 2008*, pp. 31–40, October 2008.

Zhou, Bolei, Lapedriza, Agata, Xiao, Jianxiong, Torralba, Antonio, and Oliva, Aude. Learning deep features for scene recognition using places database. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 487–495. Curran Associates, Inc., 2014.