Explanations for Multinomial Classifiers

Tips and Tricks for Practitioners

Pramit Choudhary[*]
Los Angeles, CA
pramit.choudhary@h2o.ai

Navdeep Gill[*]
Mountain View, CA
navdeep.gill@h2o.ai

Patrick Hall[†]
Washington, DC
patrick.hall@h2o.ai

Abstract

1 Introduction

This short discussion bookends popular and practical texts on machine learning explanations by Chaudhary, Gill, Hall et al by specifically addressing the common and somewhat vexing problem of explaining the behavior and predictions of multinomial classifiers [7], [6], [2].

2 Notation

To facilitate technical descriptions of explanatory techniques, notation for input and output spaces, datasets, and models is defined.

2.1 Spaces

- Input features come from the set $\mathcal{X}$ contained in a *P*-dimensional input space, $\mathcal{X} \subset \mathbb{R}^P$.
- Known labels corresponding to instances of $\mathcal{X}$ come from the set $\mathcal{Y}$ contained in a *C*-dimensional input space, $\mathcal{Y} \subset \mathbb{R}^C$.
- Learned output responses come from the set $\hat{\mathcal{Y}}$. For classification models the set $\hat{\mathcal{Y}}$ typically contains a column vector for each unique class in $\mathcal{Y}$. In this text, the space $\hat{\mathcal{Y}}$ is said to be contained in a C'-dimensional output space, $\hat{\mathcal{Y}} \subset \mathbb{R}^{C'}$.

2.2 Datasets

- The input dataset $\mathbf{X}$ is composed of observed instances of the set $\mathcal{X}$ with a corresponding dataset of labels $\mathbf{Y}$, observed instances of the set $\mathcal{Y}$.
- Each i-th observation of $\mathbf{X}$ is denoted as $\mathbf{x}^{(i)} = [x_0^{(i)}, x_1^{(i)}, \ldots, x_{P-1}^{(i)}]$, with corresponding i-th labels in $\mathbf{Y}, \mathbf{y}^{(i)} = [y_0^{(i)}, y_1^{(i)}, \ldots, y_{C-1}^{(i)}]$, and corresponding predictions in $\hat{\mathbf{Y}}, \hat{\mathbf{y}}^{(i)} = [\hat{y}_0^{(i)}, \hat{y}_1^{(i)}, \ldots, \hat{y}_{C'-1}^{(i)}]$.
- $\mathbf{X}$ and $\mathbf{Y}$ consist of N tuples of observations: $[(\mathbf{x}^{(0)}, \mathbf{y}^{(0)}), (\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \ldots, (\mathbf{x}^{(N-1)}, \mathbf{y}^{(N-1)})]$.
- Each j-th input column vector of $\mathbf{X}$ is denoted as $X_j = [x_j^{(0)}, x_j^{(1)}, \ldots, x_j^{(N-1)}]^T$.

[*]H2O.ai
[†]H2O.ai and George Washington University

Preprint. Work in progress.

2.3 Models

- A type of machine learning model g, selected from a hypothesis set $\mathcal{H}$, is trained to represent an unknown signal-generating function f observed as $\mathbf{X}$ with labels $\mathbf{Y}$ using a training algorithm $\mathcal{A}$:
- g generates learned output responses on the input dataset $g(\mathbf{X}) = \hat{\mathbf{Y}}$, and on the general input space $g(\mathcal{X}) = \hat{\mathcal{Y}}$.
- The model to be explained is denoted as g.

3 Global Analysis

3.1 Data

Subsequent sections will use simulated data that empirically demonstrates the desired relationships between input feature importance and interactions in the input space $\mathbf{X}$, the label space $f(\mathbf{X}) = \mathbf{Y}$, a GBM model to be explained g_{GBM}, and a decision tree surrogate h_{tree}. Data with a known signal-generating function depending on four input features with interactions and with eight noise features is simulated such that:

$$f = \text{num}_1 * \text{num}_4 + |\text{num}_8| * \text{num}_9^2 + e \tag{1}$$

g_{GBM} is trained: $\mathbf{X}, \mathbf{f}(\mathbf{X}) \xrightarrow{\mathcal{A}} g_{\text{GBM}}$ such that $g_{\text{GBM}} \approx f$. Then h_{tree} is extracted by $\mathbf{X}, g_{\text{GBM}}(\mathbf{X}) \xrightarrow{\mathcal{A}} h_{\text{tree}}$, such that $h_{\text{tree}}(\mathbf{X}) \approx g_{\text{GBM}}(\mathbf{X}) \approx f(\mathbf{X})$.

3.2 Decision Tree Surrogate

Given a learned function g, a set of learned output responses $g(\mathbf{X}) = \hat{\mathbf{Y}}$, and a tree splitting and pruning approach $\mathcal{A}$, a global – or over all $\mathbf{X}$ – surrogate decision tree h_{tree} can be extracted such that $h_{\text{tree}}(\mathbf{X}) \approx g(\mathbf{X})$:

$$\mathbf{X}, g(\mathbf{X}) \xrightarrow{\mathcal{A}} h_{\text{tree}} \tag{2}$$

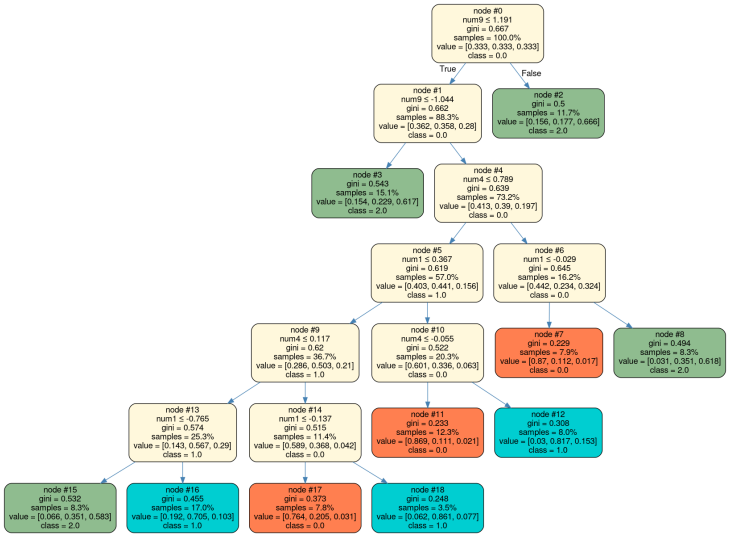

Figure 1:

Prescribed methods for training h_{tree} do exist [3] [1]. In practice, straightforward cross-validation and pruning approaches are often sufficient. Moreover, comparing cross-validated training error to traditional training error can give an indication of the stability of the single decision tree h_{tree}.

Elegantly handles high cardinality targets.

3.3 Decision Boundary Plots

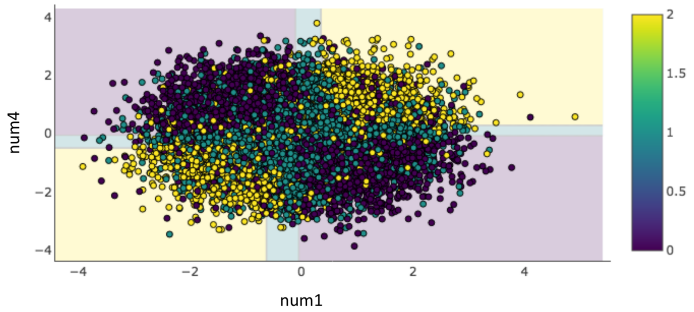

Figure 2:

What to do if very high cardinality in response, $\mathbf{Y}$:

- 2- or 3-D plot against most important variables
- 2- or 3-D plot against sparse, interpretable extracted features: NMF, Sparse PCA

3.4 Shapley Global Feature Importance

Shapley explanations, including tree shap and even certain implementations of LIME, are a class of additive, consistent local feature contribution measures with long-standing theoretical support [10]. Shapley explanations are the only possible locally accurate and consistent feature contribution values, meaning that Shapley explanation values for input features always sum to $g(\mathbf{x})$ and that Shapley explanation values can never decrease for some x_j when g is changed such that x_j truly makes a stronger contribution to $g(\mathbf{x})$ [10].

$$g(\mathbf{x}) = \phi_0 + \sum_{j=0}^{j=\mathcal{P}-1} \phi_j \mathbf{z}_j \tag{3}$$

$$\phi_j = \sum_{S \subseteq \mathcal{P} \setminus \{j\}} \frac{|S|!(\mathcal{P} - |S| - 1)!}{\mathcal{P}!} [g_x(S \cup \{j\}) - g_x(S)] \tag{4}$$

Shapley values can be estimated in different ways. Tree shap is a specific implementation of Shapley explanations. It does not rely on surrogate models. Both tree shap and a related technique known as *treeinterpreter* rely instead on traversing internal tree structures to estimate the impact of each x_j for some $g(\mathbf{x})$ of interest [9], [12].

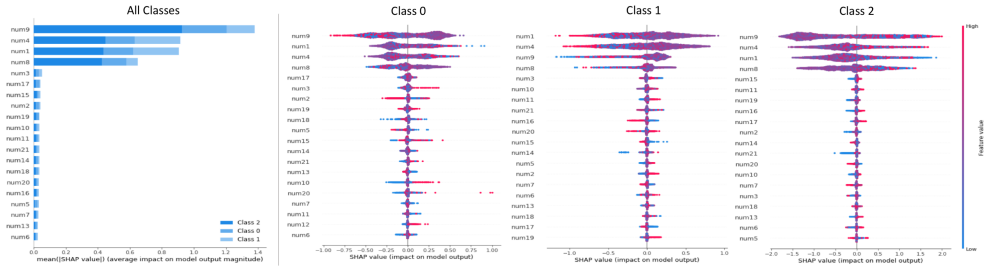

Figure 3: Shapley summary plot for known signal-generating function $f = \text{num}_1 * \text{num}_4 + |\text{num}_8| * \text{num}_9^2 + e$, and for learned GBM response function g_{GBM} per response outcome.

Simulated data is used to illustrate the utility of tree shap. Shapley explanations are estimated on $g_{\text{GBM}}(\mathbf{X})$ for a simulated test set $\mathbf{X}$ with known signal-generating function f. Results are presented in Figure 3. Firstly, the Shapley explanations are shown globally across all class outcomes in a stacked bar chart broken down by absolute global Shapley values per class outcome. This is a good way to see a overall picture of Shapley explanations for multinomial classifiers. Secondly, the Shapley explanations are broken down per class outcome in subsequent charts. All feature contributions for $\text{num}_1, \text{num}_4, \text{num}_8$ and num_9 are seen as most important across all class outcomes both in the global stacked bar chart and per class outcome. However, they are not seen in the same order. For example, class 0 and class 2 share the same ordering of $\text{num}_1, \text{num}_4, \text{num}_8$ and num_9 but class 1 does not (num_1 and num_9 are swapped). This information can be used to investigate why the Shapley explanataions differ between different class outcomes at a global level.

3.4.1 Recommendations

- Tree shap is ideal for estimating high-fidelity, consistent, and complete explanations of decision tree and decision tree ensemble models, perhaps even in regulated applications to generate regulator-mandated reason codes (also known as turn-down codes or adverse action codes).
- Because tree shap explanations are offsets from a global intercept, each ϕ_j can be interpreted as the difference in $g(\mathbf{x})$ and the average of $g(\mathbf{X})$ associated with some input feature x_j [11].
- What to do if very high cardinality in response, $\mathbf{Y}$:
  - Examine top-K most frequent classes
  - Examine top-K most accurate and inaccurate classes
  - Examine classes with highest variance in sum(absolute(shap))

3.5 Partial Dependence and ICE

Partial dependence (PD) plots are a widely-used method for describing the average predictions of a complex model g across some partition of data $\mathbf{X}$ for some interesting input feature X_j [4]. Individual conditional expectation (ICE) plots are a newer method that describes the local behavior of g for a single instance $\mathbf{x} \in \mathcal{X}$. Partial dependence and ICE can be combined in the same plot to identify interactions modeled by g and to create a holistic portrait of the predictions of a complex model for some X_j [5].

Following Friedman et al. a single feature $X_j \in \mathbf{X}$ and its complement set $\mathbf{X}_{(-j)} \in \mathbf{X}$ (where $X_j \cup \mathbf{X}_{(-j)} = \mathbf{X}$) is considered. $\text{PD}(X_j, g)$ for a given feature X_j is estimated as the average output for a particular class outcome, C', of the learned function $g(\mathbf{X})$ when all the components of X_j are set to a constant $x \in \mathcal{X}$ and $\mathbf{X}_{(-j)}$ is left unchanged. $\text{ICE}(x_j, \mathbf{x}, g)$ for a given instance $\mathbf{x}$ and feature x_j is estimated as the output for a particular class outcome, C', for $g(\mathbf{x})$ when x_j is set to a constant $x \in \mathcal{X}$ and all other features $\mathbf{x} \in \mathbf{X}_{(-j)}$ are left untouched. Partial dependence and ICE curves are usually plotted over some set of constants $x \in \mathcal{X}$.

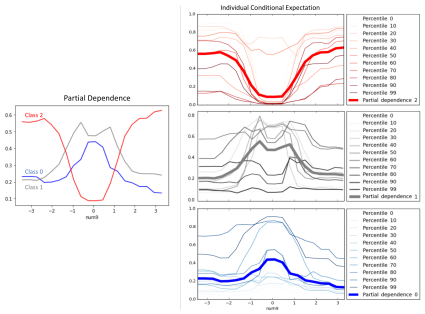

Figure 4: Partial dependence and ICE curves per response outcome for previously defined known signal-generating function f, learned GBM response function g_{GBM}, and important input feature num_9.

As in Section 3.2, simulated data is used to highlight desirable characteristics of partial dependence and ICE plots. In Figure 4 partial dependence and ICE at the minimum, maximum, and each decile of $g_{\text{GBM}}(\mathbf{X})$ are plotted per response outcome. The known quadratic behavior of num_9 is plainly visible, except for low/high value predictions across certain deciles per response outcome. When partial dependence and ICE curves diverge, this often points to an interaction that is being averaged out of the partial dependence. Given the form of Equation 1, there is a known interaction between num_9 and num_8. Combining the information from partial dependence and ICE plots with h_{tree} can help elucidate more detailed information about modeled interactions in g.

3.5.1 Recommendations

- Combining h_{tree} with partial dependence and ICE curves per class outcome is a convenient method for detecting, confirming, and understanding important interactions in g.
- What to do if very high cardinality in response, $\mathbf{Y}$:
  - Examine top-K most frequent classes
  - Examine top-K most accurate and inaccurate classes
  - Examine classes with highest variance in partial dependence
  - Examine classes with largest differences between partial dependence and ICE

4 Supplementary Materials

UCI credit card dataset [8].

`https://github.com/navdeep-G/interpretable-ml/tree/master/notebooks`

5 Conclusion

6 NIPS Style examples

Paragraphs There is also a `\paragraph` command available, which sets the heading in bold, flush left, and inline with the text, with the heading followed by 1 em of space.

6.1 Citations, figures, tables, references

These instructions apply to everyone.

6.2 Citations within the text

The `natbib` package will be loaded for you by default. Citations may be author/year or numeric, as long as you maintain internal consistency. As to the format of the references themselves, any style is acceptable as long as it is used consistently.

The documentation for `natbib` may be found at

`http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf`

Of note is the command `\citet`, which produces citations appropriate for use in inline text. For example,

```
\citet{hasselmo} investigated\dots
```

produces

Hasselmo, et al. (1995) investigated…

If you wish to load the `natbib` package with options, you may add the following before loading the `nips_2018` package:

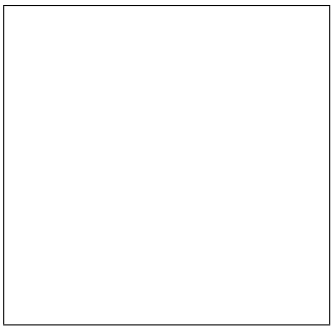

Figure 5: Sample figure caption.

```
\PassOptionsToPackage{options}{natbib}
```

If `natbib` clashes with another package you load, you can add the optional argument `nonatbib` when loading the style file:

```
\usepackage[nonatbib]{nips_2018}
```

As submission is double blind, refer to your own published work in the third person. That is, use "In the previous work of Jones et al. [4]," not "In our previous work [4]." If you cite your other papers that are not widely available (e.g., a journal paper under review), use anonymous author names in the citation, e.g., an author of the form "A. Anonymous."

New preprint option for 2018 If you wish to post a preprint of your work online, e.g., on arXiv, using the NIPS style, please use the `preprint` option. This will create a nonanonymized version of your work with the text "Preprint. Work in progress." in the footer. This version may be distributed as you see fit. Please **do not** use the `final` option, which should **only** be used for papers accepted to NIPS.

At submission time, please omit the `final` and `preprint` options. This will anonymize your submission and add line numbers to aid review. Please do *not* refer to these line numbers in your paper as they will be removed during generation of camera-ready copies.

The file `nips_2018.tex` may be used as a "shell" for writing your paper. All you have to do is replace the author, title, abstract, and text of the paper with your own.

The formatting instructions contained in these style files are summarized in Sections **??**, **??**, and 6.1 below.

6.3 Footnotes

Footnotes should be used sparingly. If you do require a footnote, indicate footnotes with a number[3] in the text. Place the footnotes at the bottom of the page on which they appear. Precede the footnote with a horizontal rule of 2 inches (12 picas).

Note that footnotes are properly typeset *after* punctuation marks.[4]

6.4 Figures

6.5 Tables

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

Note that publication-quality tables *do not contain vertical rules.* We strongly suggest the use of the `booktabs` package, which allows for typesetting high-quality, professional tables:

[3] Sample of the first footnote.

[4] As in this example.

Table 1: Sample table title

	Part	
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

`https://www.ctan.org/pkg/booktabs`

This package was used to typeset Table 1.

The `\bbold` package almost always uses bitmap fonts. You should use the equivalent AMS Fonts:

```
\usepackage{amsfonts}
```

followed by, e.g., `\mathbb{R}`, `\mathbb{N}`, or `\mathbb{C}` for $\mathbb{R}$, $\mathbb{N}$ or $\mathbb{C}$. You can also use the following workaround for reals, natural and complex:

```
\newcommand{\RR}{I\!\!R} %real numbers
\newcommand{\Nat}{I\!\!N} %natural numbers
\newcommand{\CC}{I\!\!\!\!C} %complex numbers
```

Note that `amsfonts` is automatically loaded by the `amssymb` package.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

References

[1] Osbert Bastani, Carolyn Kim, and Hamsa Bastani. Interpreting Blackbox Models via Model Extraction. *arXiv preprint arXiv:1705.08504*, 2017. URL: `https://arxiv.org/pdf/1705.08504.pdf`.

[2] Pramit Choudhary. Interpreting Predictive Models with Skater: Unboxing Model Opacity. *O'Reilly Ideas*, 2018. URL: `https://www.oreilly.com/ideas/interpreting-predictive-models-with-skater-unboxing-model-opacity`.

[3] Mark W. Craven and Jude W. Shavlik. Extracting Tree-structured Representations of Trained Networks. *Advances in Neural Information Processing Systems*, 1996. URL: `http://papers.nips.cc/paper/1152-extracting-tree-structured-representations-of-trained-networks.pdf`.

[4] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. ***The Elements of Statistical Learning***. Springer, New York, 2001. URL: `https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf`.

[5] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation. *Journal of Computational and Graphical Statistics*, 24(1), 2015. URL: `https://arxiv.org/pdf/1309.6392.pdf`.

[6] Patrick Hall. On the Art and Science of Machine Learning Explanations. *arXiv preprint arXiv:1810.02909*, 2018. URL: `https://arxiv.org/pdf/1810.02909.pdf`.

[7] Patrick Hall, Wen Phan, and Sri Satish Ambati. Ideas on Interpreting Machine Learning. *O'Reilly Ideas*, 2017. URL: `https://www.oreilly.com/ideas/ideas-on-interpreting-machine-learning`.

[8] M. Lichman. UCI Machine Learning Repository, 2013. URL: http://archive.ics.uci.edu/ml.

[9] Scott M. Lundberg, Gabriel G. Erion, and Su-In Lee. Consistent Individualized Feature Attribution for Tree Ensembles. *arXiv preprint arXiv:1802.03888*, 2018. URL: https://arxiv.org/pdf/1706.06060.pdf.

[10] Scott M. Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[11] Christoph Molnar. *Interpretable Machine Learning*. christophm.github.io/interpretable-ml-book, 2018. URL: https://christophm.github.io/interpretable-ml-book/.

[12] Ando Saabas. Interpreting Random Forests, 2014. URL: http://blog.datadive.net/interpreting-random-forests/.