Received 26 April 2016; | Revised 6 June 2016; | Accepted 6 June 2016

DOI: xxx/xxxx

ARTICLE TYPE

Visual Diagnostics of a Model Explainer – Tools for the Assessment of LIME Explanations

Katherine Goode*[1] | Heike Hofmann[1,2]

[1]Department of Statistics, Iowa State University, Iowa, United States
[2]Center for Statistics and Applications in Forensic Evidence (CSAFE), Iowa State University, Iowa, United States

Correspondence
*Corresponding author. Email: kgoode@iastate.edu

Summary

The importance of providing explanations for predictions made by black-box models has led to the development of model explainer methods such as LIME (local interpretable model-agnostic explanations) [23]. LIME uses a surrogate model to explain the relationship between predictor variables and predictions from a black-box model in a local region around a prediction of interest. However, the quality of the resulting explanations relies on how well the explainer model captures the black-box model in a specified local region. Here we introduce three visual diagnostics to assess the quality of LIME explanations: (1) explanation scatterplot, (2) assessment metric plot, and (3) feature heatmap. We apply the visual diagnostics to a forensics bullet matching dataset to show examples where LIME explanations depend on the tuning parameters and the explainer model oversimplifies the black-box model. Our examples raise concerns about claims made about LIME [23] that are similar to other criticisms in the literature (Alvarez-Melis and Jaakkola [1], Laugel et al. [16], and Molnar [20]).

KEYWORDS:
LIME, black-box models, interpretability, diagnostics

1 | INTRODUCTION

In the field of statistics, there are two main uses for models: inference and prediction. Machine learning models are often used for the latter purpose. These models have proven to perform well in a wide range of prediction problems, but the accuracy of many machine learning models comes at the cost of interpretability due to their algorithmic complexity (hence the phrase "black-box models"). Model interpretability allows for the understanding and assessment of how a model produces predictions. The lack of the ability to understand and assess a model makes it difficult to trust the model, especially in areas with high stakes decisions such as health care and forensics science. The increased use of machine learning models in applications and the introduction of the General Data Protection Regulation (GDPR) in 2018 [9] has resulted in a dramatic increase in explainable machine learning research, which focuses on developing ways to explain output from machine learning algorithms.

Throughout this paper, we distinguish between interpretability and explanability of models. We define *interpretability* as the ability to directly use model parameters to understand the relationships in the data captured by the model: e.g., a linear model coefficient associated with a predictor variable indicates the amount the response variable changes based on a change in the predictor variable. In contrast, define *explanability* as the ability to use the model in an indirect manner to understand the relationships in the data captured by the model: e.g., partial dependence plots depict the marginal relationship between model predictions and predictor variables [6].

Numerous methods have been proposed to provide explanations for black-box model predictions [7, 11, 18, 19, 20]. Some are specific to one type of model (e.g. [26] and [29]), and others are model-agnostic (e.g. [5] and [30]). In this paper, we focus on one specific model-agnostic method: LIME Ribeiro et al. [23] introduced by Ribeiro, Singh, and Guestrin in 2016.

LIME (local interpretable model-agnostic explanations) is a method that uses a surrogate model to relate predictor variables to black-box model predictions (i.e. a model explainer) [23]. We distinguish between the terms of model explainer and explainer model: by *model explainer* we denote the method for explaining a complex model using a surrogate model, while the *explainer model*, or simply the *explainer*, is the surrogate model.

While some model explainers focus on understanding a model at the global level, LIME claims to provide explanations for individual predictions (local). Additionally, LIME is designed to work with any model (model-agnostic) and to produce easily understandable results (explanations) [23]. Conceptually, LIME fits a simple (interpretable) model, the explainer model, meant to capture the behavior of the (complex) black-box model in a local region around a prediction of interest. The simple model then provides interpretable estimates for variables that most influenced the prediction made by the complex model.

Figure 1 provides a visualization of this conceptual understanding of LIME. The two plots show the predictions from a hypothetical black-box model plotted against the two predictor variables used to fit the hypothetical model. The diamond shaped points represent a prediction of interest. A Gower distance metric [10] is used to define locality such that the size of the points represent the inverse of the Gower distance raised to some value and indicate the proximity to the prediction of interest. In the example of Figure 1 an exponent of 50 is used to emphasize a very local region around the prediction of interest. A ridge regression model weighted by the proximity values is used as the explainer model with the black-box predictions as the response variable and standardized versions of two features in the data as predictor variables. Standardized features allow direct comparisons of the model coefficients. The explainer model is depicted by the black lines in the figure.

The plot on the left of Figure 1 shows the relationship between the black-box predictions and Feature 1. Here, the explainer model is plotted with Feature 2 fixed at the observed value of the prediction of interest. The explainer model captures the relationship in the immediate neighborhood around the prediction of interest with a slope of 0.068. The plot on the right shows no global or local relationship between the black-box predictions and Feature 2. Here, the explainer model is plotted with Feature 1 set as the observed value of the prediction of interest, and it has an appropriately small slope of -0.001. The magnitude of the slope associated with Feature 1 is larger than the slope of Feature 2, which suggests that Feature 1 plays a more important role in the prediction made by the black-box model for the prediction of interest. This explanation agrees with a visual assessment of the relationships between the predictions and predictor variables.

The concept of LIME is relatively simple: use an interpretable model to approximate a complex model in a local region. However, a practical implementation of LIME is not straightforward, and research is being done to improve the procedure [16]. The current implementations of LIME[1] [21] offer various tuning parameters (see Section 2) that affect the explainer model and ultimately, the explanation. Since the explainer model is an approximation of the complex model and not a direct interpretation, the explanations produced by an explainer model are subject to the quality of the approximation. Thus, in order to achieve accurate explanations, the tuning parameters selected need to be assessed.

We consider an example where the choice of exponent for the weights is crucial to the quality of the explanation. The plots in Figure 2 show the same data as Figure 1 , but the Gower distance metric exponent is decreased to 1 (the default exponent in the *lime* R package). This causes the observations that are further away from the prediction of interest to be given larger weights than before. In Figure 1 , the explainer model captures the relationship between the black-box predictions in the immediate neighborhood of the prediction of interest. This cannot be said of the explainer model in Figure 2 . In addition, the magnitude of the slope (0.011) associated with Feature 2 is larger than that of Feature 1 (0.005). Thus, this explainer model actually provides a misleading explanation that Feature 2 plays a more important role in predicting the observation of interest.

Several sources in the literature discuss the performance of LIME. One of the biggest difficulties with LIME is determining how to specify a local region [16] [20]. This is due to an unclear definition of a "local region" and how to apply LIME to achieve an appropriate local region as demonstrated by Figure 2 . Alvarez-Melis and Jaakkola [1] raise a concern pertaining to the robustness of explanations from LIME and other model explainers: they find that even small changes in predictor variables can lead to very different LIME explanations. Additionally, Ribeiro et al. [23] acknowledge that if a linear model is used as the explainer, LIME relies on a linear approximation of the explainer model to the complex model and state “if the underlying model is highly non-linear even in the locality of the prediction, there may not be a faithful explanation".

As a result of the various ways LIME can fail, it is important to assess LIME explanations. We suggest the use of visual diagnostics for assessment. In this paper, we lay out the set of claims about LIME made by Ribeiro et al. [23] and propose three visualizations for the assessment of these claims: (1) *explanation scatterplot*, (2) *feature heatmap*, (3) *assessment metric plot*. While LIME is implemented for image, tabular,

[1] https://github.com/marcotcr/lime

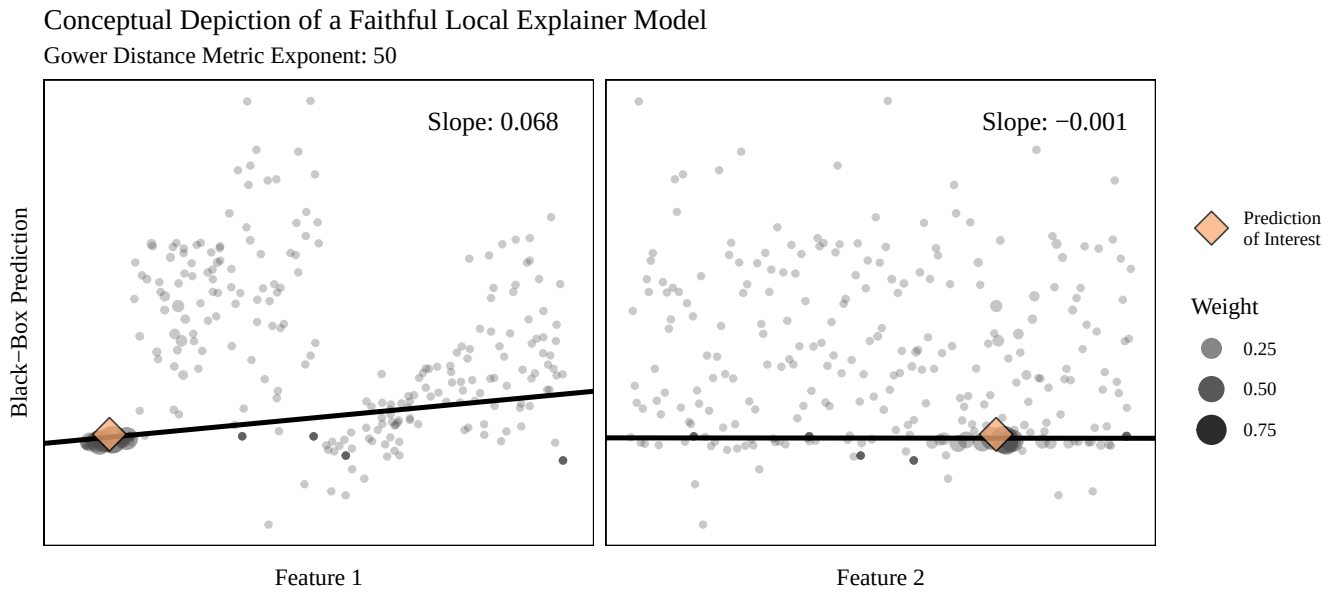

FIGURE 1 A conceptual depiction of a faithful LIME explainer model in the immediate neighborhood of a prediction of interest. The predictions from a hypothetical black-box model are plotted against the standardized values of the two hypothetical predictor variables. The diamond shaped points represent the location of a prediction of interest. The size and opacity of the circular points indicate the weight assigned based on the distance to the prediction of interest computed using the inverse of the Gower distance metric raised to the power of 50. The black lines are a weighted ridge regression model used as an explainer model that reasonably captures the relationship between the black-box predictions and the features in a local region around the prediction of interest. That is, the explainer is faithful to the complex model and produces a reasonable explanation that Feature 1 plays a more important role in the prediction of interest than Feature 2 since the magnitude of the slope associated with Feature 1 is larger.

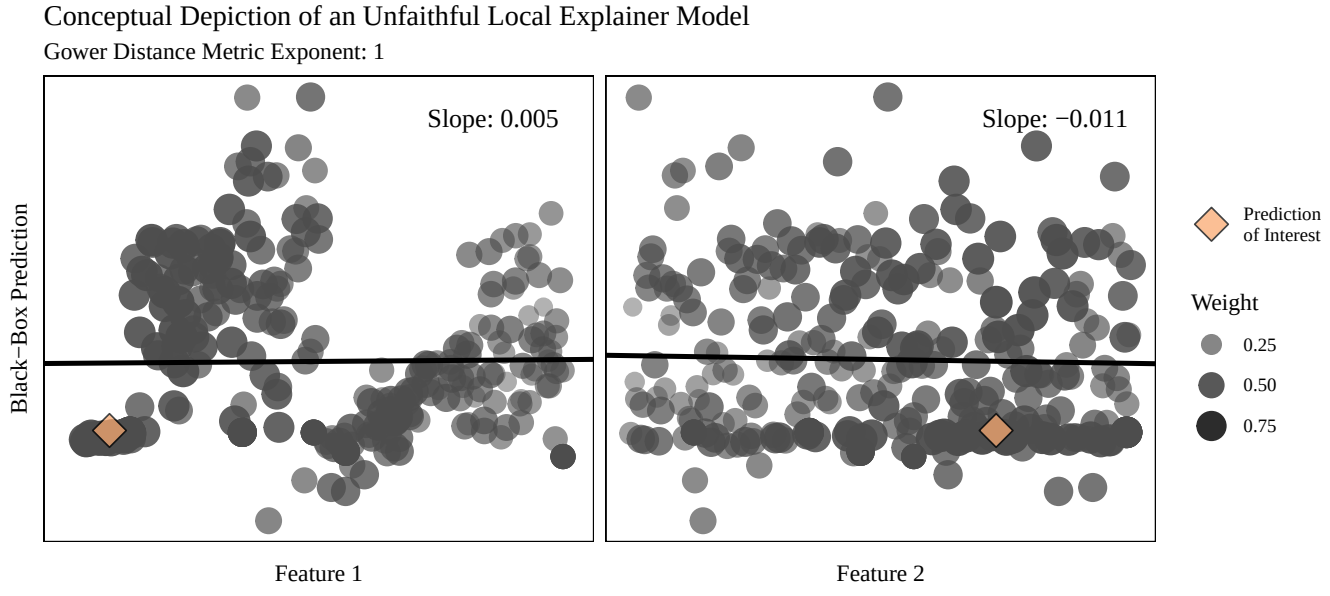

FIGURE 2 A conceptual depiction of an unfaithful local explainer model in the immediate neighborhood of a prediction of interest. A ridge regression model is fit to the same hypothetical black-box model predictions and predictor variables as in Figure 1 , but the weights are computed using a Gower exponent of 1. The explainer model is unfaithful to the complex model in the immediate neighborhood of the prediction of interest.

and text data, we only focus on tabular data. For additional simplicity, we only discuss classification prediction models with a dichotomous response variable and continuous predictor variables. However, the proposed diagnostics may be extended to a wider range of situations.

The remainder of the paper is structured as follows. Section 2 provides background and claims made by Ribeiro et al. [23] about LIME. We introduce the suggested diagnostic plots in Section 3. Then in Section 4, we demonstrate the use of the diagnostics to assess LIME explanations for a random forest model fit to a forensics bullet matching dataset. Section 5 concludes with a discussion on extensions and limitations of the diagnostic plots and concerns about LIME in regards to the claims made by Ribeiro et al. [23] brought about by the visualization examples in this paper that agree with Alvarez-Melis and Jaakkola [1], Laugel et al. [16], and Molnar [20]. The diagnostic plots included in this paper are created using the R package *limeaid* [8].

2 | BACKGROUND ON LIME

LIME was introduced in 2016 by Ribeiro et al. [23]. The authors provide an implementation of LIME in a Python package[2]. An adaption of the Python package in R has been implemented and made available by Thomas Lin-Pedersen [21]. The discussion of parameter choices and implementation details of LIME in this paper are based on the R package.

The general form of the LIME algorithm can be divided into three steps (see also [16]):

1. *Data Simulation and Interpretable Transformation*: Simulate a dataset from the original data used to fit the black-box model. Apply a transformation to the simulated data and the prediction of interest that will allow for interpretable explanations.

2. *Explainer Model Fitting*: Apply the black-box model to the simulated data to obtain predictions. Compute the distance between each of the simulated observations and the prediction of interest. Perform feature selection. Fit an interpretable model with the black-box predictions from the simulated data as the response, the selected features from the transformed simulated data as the predictors, and the distances as weights. This model is the explainer model.

3. *Explainer Model Interpretation*: Interpret the explainer model to determine which features played the most important role in the prediction of interest.

[2]https://github.com/marcotcr/lime

During the application of LIME, the user is asked to select various tuning parameter options: the number of features to return in the explanation, the simulation method, the feature selection method, and how the weights are computed. An overview of the options available for the tuning parameters is included in Appendix A.

In the original paper, Ribeiro et al. [23] make the following set of claims regarding the performance of LIME:

- *Interpretability*: The explainer model can be easily interpreted to provide meaningful explanations.
- *Faithfulness*: The explainer model sufficiently captures the relationship between the complex model predictions and the features in the local region around a prediction of interest to produce explanations that are faithful to the complex model.
- *Linearity*: By using a ridge regression model as the explainer model, it is assumed that there is a linear relationship between complex model predictions and the features in the local region around a prediction of interest.
- *Localness*: The explanations produced by LIME are local in regards to a prediction of interest.

The assumption of interpretability only depends on the complexity of the model used as explainer model. If the model is too complex to provide meaningful explanations (e.g. there are too many variables in the model), it is clear that the assumption of interpretability is violated. The other three assumptions are not as easy to assess – for those we suggest the use of diagnostic plots.

3 | VISUAL DIAGNOSTICS FOR LIME

In this section, we introduce three visual diagnostic plots for the assessment of LIME. The plots focus on different levels of application of LIME (e.g. on explanation versus a set of explanations) to assess the LIME claims from different perspectives:

1. *Explanation Scatterplot* (Section 3.2): Comparison of the explainer and complex models for an individual prediction of interest.

2. *Feature Heatmap* (Section 3.3): Comparison of features selected by LIME across applications of LIME with different tuning parameters.

3. *Assessment Metric Plot* (Section 3.4): Comparison of performance metrics for LIME across applications of LIME with different tuning parameters.

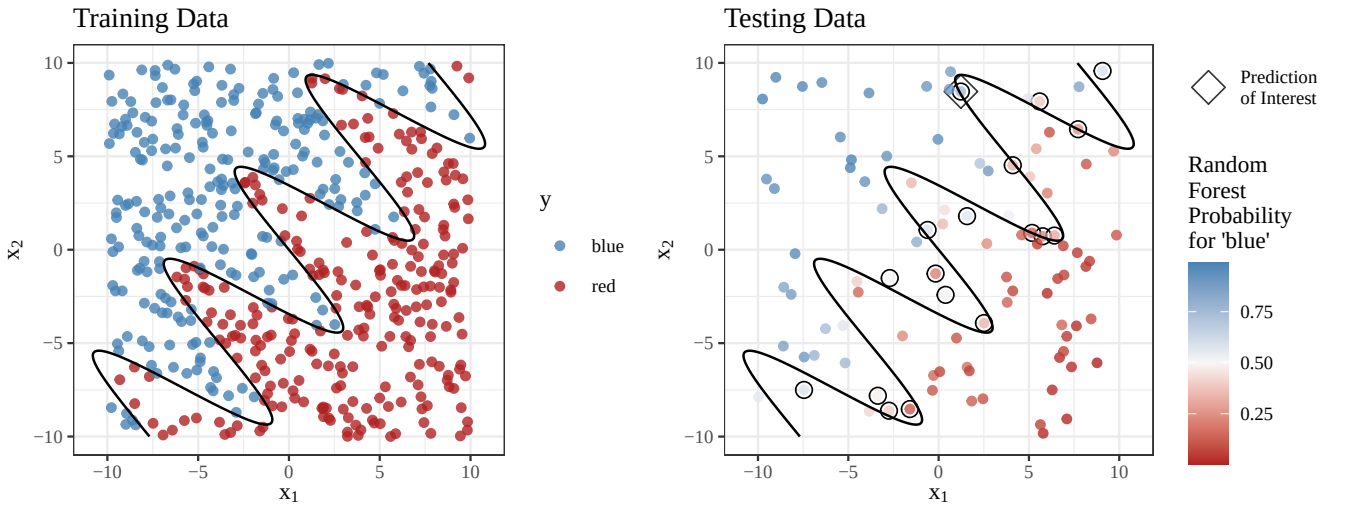

FIGURE 3 Plots of x_2 versus x_1 from the training (left) and testing (right) sets of the sine data introduced in Section 3. The true classification boundary is shown as the solid black line in both plots. The color of the training data points represents the value of the observed response variable (y). The color of the testing data points represents the random forest probability that an observation belongs to the category of blue ($\hat{y}$). The 18 cases that are misclassified by the random forest are identified by black circles. The prediction of interest to explain is indicated by a diamond.

The sine data

To demonstrate the visualizations, we generate an example dataset that will be referred to as the sine data. The sine data contains 600 observations with three features and one response variable. The features, x_1, x_2, and x_3, are randomly sampled from Unif(-10, 10), Unif(-10, 10), and N(0, 1) distributions, respectively. A binary response variable y is created using a rotated sine curve. In particular, let $x_1' = x_1\cos(\theta) - x_2\sin(\theta)$ and $x_2' = x_1\sin(\theta) + x_2\cos(\theta)$ where $\theta = -0.9$. Then y is defined as

$$y = \begin{cases} \text{blue} & \text{if } x_2' > \sin\left(x_1'\right) \\ \text{red} & \text{if } x_2' \leq \sin\left(x_1'\right) \end{cases}. \quad (1)$$

Note that due to the creation of y in this manner, y is dependent on x_1 and x_2 and independent of x_3. From the plots in **??** we see that the global relationship between response y and features x_1 and x_2 is linear: the probability for label blue increases with the difference between features x_2 and x_1. Locally, the relationship between y and features x_2 and x_1 varies a lot more around the line of identity. Here, the relationship is determined by the sine waves. However, the sine is a good-natured function that can be approximated well linearly.

The dataset is divided into a training set of 500 observations and a testing set of 100 observations. A random forest model is fit using the R package *randomForest* (version 4.6.14) [17] with the default settings. The model is applied to the test set to obtain predictions. Figure 3 shows scatterplots of x_2 versus x_1 from the training data (left) and the testing data (right). Both plots include the true classification boundary of the rotated sine function plotted as the solid black line. The training data are colored by the observed response variable (y), and the testing data are colored by the prediction probabilities from the random forest model ($\hat{y}$). The random forest model misclassifies 18 points on the classification boundary. These are identified by circles in Figure 3 .

For the presentation of the explanation scatterplot, we focus on the misclassified point with (x_1, x_2, x_3) coordinates of (1.23, 8.47, -0.99) indicated by a diamond in Figure 3 . Misclassified points are often of interest to explain since they may provide information about ways to improve the model. For the introduction of the other three plots, we use all observations in the sine datatest data as points of interest.

All runs of LIME in this paper are executed in a forked version[3] of the development version of the R package *lime* (0.5.1) [21]. The forked version is functionally indistinguishable from Pedersen's implementation but it allows us to export internal values relevant for an assessment of the explainer. The R package *limeaid* [4] (0.0.1) implements diagnostic plots based on the output of the altered version of *lime*.

A quantile bin based simulation method is used for five of the LIME applications with the number of bins varying from 2 to 6 by application. We use 6 bins as the maximum, because the complexity of the explanations increases with the number of bins. The sixth application of LIME uses a kernel density

[3] https://github.com/goodekat/lime
[4] Available on GitHub at https://github.com/goodekat/limeaid

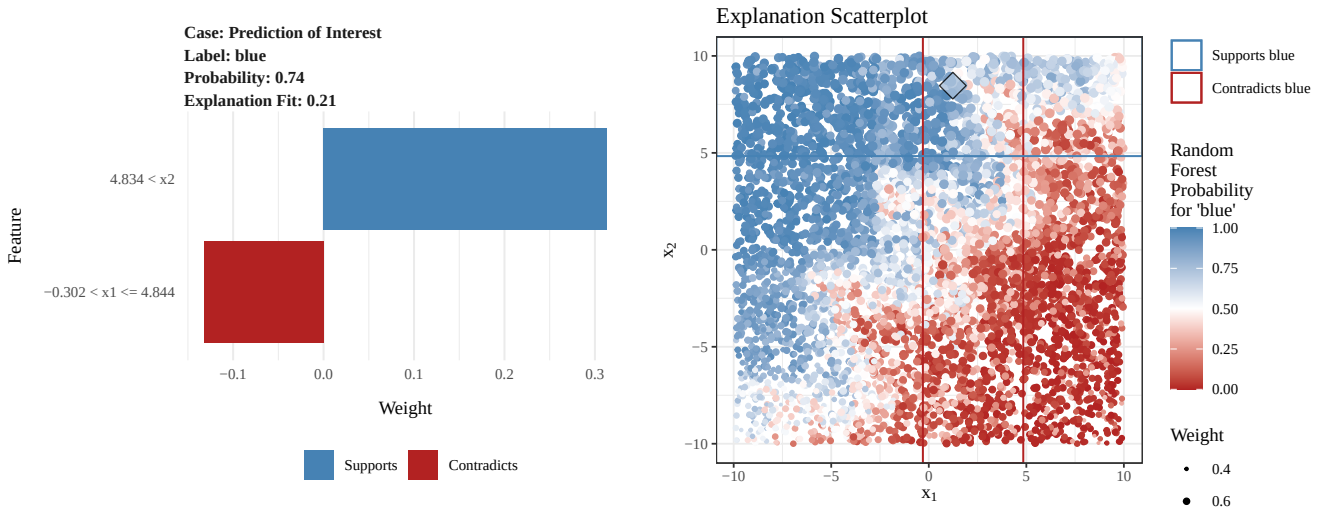

FIGURE 4 (left) Visualization from the *lime* R package of the LIME explanation for the sine data prediction of interest. The bars represent the explainer model coefficients. (right) *Explanation Scatterplot*. The points represent the simulated data colored by the random forest probabilities and sized by the assigned LIME weights. XXX could you make the complex prediction **much** lighter for this example? The diamond and the lines are hard to see, but should be the focus. The prediction of interest is shown as the diamond shaped point. The explainer model indicator variables associated with x_1 and x_2 (quantile bins containing the prediction of interest) are depicted by the solid lines. The line colors represent the explainer model coefficient signs and indicate whether the feature supports a prediction of 'blue' (blue lines) or supports a prediction of 'red' (red lines). This figure shows that the quantile bins are not flexible enough to capture the relationship between the random forest predictions and the x_1 and x_2 values.

simulation method. The default methods for feature selection (forward selection) and the computation of the weights (Gower distance raised to an exponent of 1) are used for all applications.

3.1 | A LIME explanation

XXX I've just pulled the description of the left hand side of figure 4 out of the next section. To demonstrate a LIME explanation, we focus on the misclassified point indicated in Figure 3 by the diamond as the prediction of interest. The left plot in Figure 4 shows a visualization of the explanation for the prediction of interest as suggested by Ribeiro et al and implemented in the *lime* R package. The figure shows that the random forest returned a probability of 0.74 that the observation of interest belongs to category blue (denoted as the "label" in the figure).The lengths of the bars represent the explainer model coefficients associated with the indicator variables chosen by LIME as important. The color of the bar denotes the sign of the coefficient and whether the feature supports or contradicts a random forest classification of 'blue'. The "explanation fit" value is the deviance ratio from the R package *glmnet* [25] for a ridge regression model. In other words, this is the R^2 value associated with the explainer model. In this case, it is 0.21 suggesting that the explainer model is not a good linear fit. However, it is commonly accepted that R^2 has limitations for assessing the quality of fit of a model [24], and it should not be used as the only metric in a model assessment.

The explanation for the prediction of interest depicted in Figure 4 can be interpreted as follows. A random forest classification of blue is supported by the prediction of interest having a value of x_2 that is greater than 4.834, but the prediction of interest having a value of x_1 that is greater than -0.302 and less than or equal to 4.844 provides support against a classification of blue. Because the support for blue by the value of feature x_1 outweighs the support for red by the value of feature x_2, the overall LIME explanation favors a label of blue for the prediction of interest. This explanation agrees with the random forest probability of 0.744 for blue. Note that both models classify the observation incorrectly. Based on the plot on the left of Figure 4 alone, it is not possible to make an informed assessment of the explanation.

3.2 | Explanation Scatterplots

XXX I think we could reduce some of the example for the specific example a bit, now that the LIME explanation is moved out, there is not that much need to repeat the general ideas.

For a further assessment of the explainer model, we turn to an *explanation scatterplot*: the *explanation scatterplot* is a visual diagnostic for assessing the LIME claims of locality and fidelity for an individual explanation by juxtaposing the complex and explainer models in one plot. The format of an explanation scatterplot depends on the LIME simulation method. We introduce the explanation scatterplot here under the *lime* R package default method in which the data are simulated uniformly from four quantile bins. In this scenario, LIME converts continuous predictor variables to indicator variables identifying whether the variable value falls in the same quantile bin as the prediction of interest or not. The indicator variables are used as the explainer model features.

The explanation scatterplot is built by plotting the LIME simulated data for the top two features identified by the explanation in a scatterplot and coloring these points by the predictions from the complex model. The size of the points represents the weight assigned by LIME. In order to show the LIME results for the observation of interest, lines are drawn on top of the points. These lines represent the boundaries of the indicator variables used to fit the explainer model. The color of the lines denote whether LIME indicates that a feature supports or contradicts a class prediction. Appendix B addresses the explanation scatterplot formats in other LIME simulation scenarios.

An explanation scatterplot for the example of the previous section is shown on the right hand side of Figure 4 . The points are the simulated values of x_2 versus x_1 colored by the random forest predictions and sized by the proximity weight used to fit the ridge regression explainer model. The explainer model is represented by the solid horizontal and vertical lines identifying the boundaries of the quantiles that contain the prediction of interest. The color of the lines indicate whether the explainer model coefficient is positive (blue) or negative (red). Thus, the x_2 coefficient is positive, so the x_2 value of the prediction of interest supports a random forest prediction of 'blue', and the x_1 coefficient is negative, so the x_1 value of the prediction of interest contradicts a random forest prediction of 'blue'. The prediction of interest is represented by the diamond shaped point.

By juxtaposing the random forest predictions and the explainer model boundaries, we are able to assess the faithfulness and localness of the explainer model. First consider the claim of localness. The weights remain relatively high outside of the intersection of the two quantile bins suggesting that the LIME explanation is highly influenced by points outside of the bin containing the prediction of interest. While it is still difficult to say whether or not the claim of localness has been violated, it is possible to say that the weights assigned to the simulated data do not agree with the local region assigned by the intersection of the quantile bins.

Now, consider the claim of faithfulness. Within the intersection of the x_1 and x_2 quantile bins, there are more random forest probabilities above 0.5. This indicates why the magnitude of the coefficient associated with x_2 is larger than that of x_1. However, the pattern of the random forest predictions in this plot suggest that better explanations for this prediction exist. One example of a better explanation is to say that the prediction of interest received a random forest prediction greater than 0.5 since the observation of interest falls in the region where x_1 is less than 2.5 and x_2 is greater 4.5. A more localized explanation is to say that the region around the prediction of interest shows that observations with $-0.3 < x_1 <= 2.5$ and $4.8 < x_2 <= 8.75$ are all assigned probabilities greater than 0.5. The procedure implemented by LIME is not flexible enough to capture either of these explanations.

The explanation scatterplot shows issues with a definition of locality and an oversimplification of the random forest model. LIME does provide an explanation for the prediction of interest, which makes sense based on the quantile bins used. However, the explanation scatterplot identifies that better explanations for the prediction of interest exist. It is difficult to assess linearity from an explanation scatterplot, but a residual plot of the explainer model could be used to check the linearity claim. See Appendix C for the residual plot associated with the explanation considered in Figure 4 and shows a violation of the linearity claim.

The sine data explanation only includes two features. In situations where more than two features are included in a LIME explanation, the explanation scatterplots can be extended to a generalized pairs plot [4] that includes all pairwise combinations of features. Generalized pairs plots (and scatterplot matrices in general) have diminishing value as the number of features increase [14] [27]. Machine learning models are commonly fit using a large number of features, and therefore, a generalized pairs plot of explanation scatterplots for all features would be ineffective. However, when applying LIME, the user selects the number of features to return in the explanation. In the *lime* R package, Pedersen and Benesty [21] encourage users to select less than 10 features to include in the explanation. As a result, specifying a small number of features to include in a LIME explanation makes it feasible to use generalized pairs plot of explanation scatterplots.

3.3 | Feature Heatmap

Explanations produced by LIME are likely to be affected by the choice of tuning parameters. An example of this is shown by Figures 1 and 2 where the method used to weight the observations influenced the explanation. As of the time of writing this manuscript, we have encountered no recommendations for how to determine which method to use besides for the default

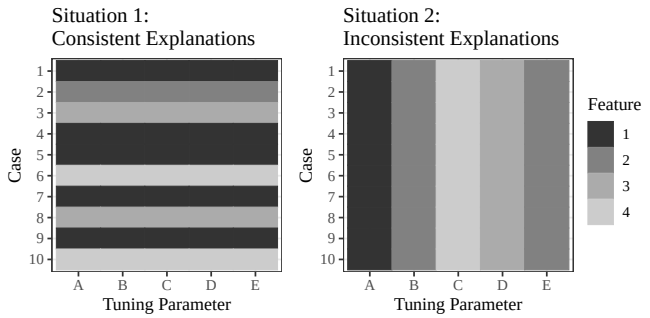

FIGURE 5 Hypothetical examples of feature heatmaps in two possible situations. The heatmaps show the top feature chosen for 10 cases across 5 different sets of LIME tuning parameters. The color of the cell indicates the feature chosen by LIME.

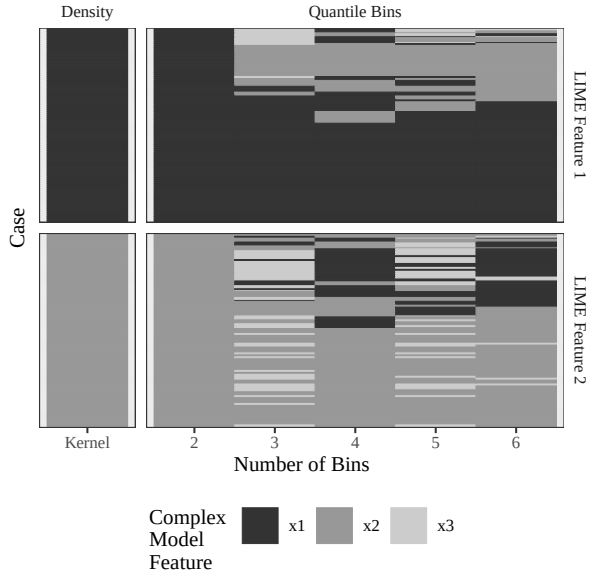

FIGURE 6 *Feature Heatmap*. An example feature heatmap of the explanations from the applications of LIME with different tuning parameters to the sine data. The cases from the test set are plotted on the y-axis, and the tuning parameter values (simulation methods here) are included on the x-axis. The colors of the tiles indicate the feature selected by LIME for the corresponding case and tuning parameter value. The plot is faceted by the first features selected by LIME (top facet) and the second most important features (bottom facet). The column facets separate the kernel density method from the quantile bin methods. The vertical striping indicates that the LIME explanations are not consistent across tuning parameters.

settings in *lime* (confirm this). In order to compare the explanations produced by LIME using different tuning parameters, we visualize an overview the explanations in the *feature heatmap* diagnostic plot.

The feature heatmap uses colors to identify the features selected by LIME across tuning parameters for sets of explanations (referred to as cases here) faceted by the position of feature importance assigned by LIME. That is, for LIME applied with t sets of tuning parameters to n cases to select the f top features, create f heatmaps (one for each of the positions of importance determined by the magnitude of the explainer model coefficients) with the cases on the y-axis, the tuning parameters on the x-axis, and the cells colored by the feature chosen for the corresponding case and tuning parameter. Additional tuning parameters may also be included in the plot via facets. This plot is used assess the locality of the explanations and to compare results from different input options.

Two hypothetical examples of feature heatmaps are included in Figure 5 . The plots were created with the assumption that LIME is applied to select the top feature out of $p = 8$ features for $n = 10$ cases with $t = 5$ sets of tuning parameters. Situation 1 shows an example where the features selected are consistent across tuning parameters within a case but vary across cases within a tuning parameter. This is the ideal situation, because the LIME explanations do not depend on the tuning parameters but do depend on the location of the observation in the feature space. Situation 2 shows an example where the features selected vary across tuning parameters within a case but are consistent across cases within a tuning parameter. This situation indicates that the features selected by LIME are dependent on the tuning parameters, and the explanations are not local because the same feature is chosen regardless of the case. In practice, it is expected that the plot will exhibit a combination of these two situations.

Figure 6 shows the feature heatmap for the LIME applications to the 100 observations in the sine data test set. The most important (top facet) and second most important (bottom facet) features are shown in this plot. For the quantile bins, the original features prior to the indicator variable transformation are included since it is obvious that different features would be selected when the sizes of the bins change. This figure shows that for kernel density and 2 quantile bins, LIME selects x_1 as the most important feature and x_2 as the second most important feature across all cases in the test set. These explanations are not local. There is variability in the features selected by LIME for 3 to 6 quantile bins. For these tuning parameters, there is a mix of horizontal and vertical stripes, which suggests a dependence on tuning parameters. Note that the explanations from 3 and 5 quantile bins include the selection of the random noise variable (x_3) as an important variable in many predictions, which should not be the case. The pattern seen in the

explanations for 3 to 6 quantile bins may suggest local explanations, but the dependence on tuning parameter makes it unclear which set of explanations to use.

3.4 | Assessment Metric Plot

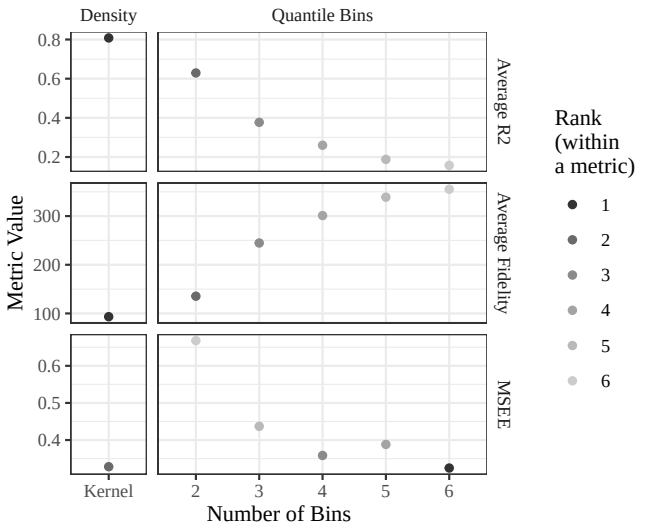

FIGURE 7 *Assessment metric plot*. The assessment metrics computed on the applications of LIME to the sine data test set are included in this example assessment metric plot. Each horizontal facet corresponds to one of three metrics: average R^2, average fidelity, or MSEE. The LIME tuning parameter (simulation methods in this example) is plotted on the x-axis, and the metric values are shown on the y-axis. The points are colored by rank (within a metric) indicating best to worst (dark to light). The kernel density simulation methods performs well according to all three metrics.

The feature heatmap for the sine data in the previous section shows an example of inconsistent LIME explanations across tuning parameters. In this situation, the user must determine which set of explanations to trust. One way to do this is to compute assessment metrics for each set of explanations to identify the optimal tuning parameters. We discuss three metrics for this purpose and present a visual comparison in an *assessment metric plot*.

Each metric presented below is computed on a set of LIME explanations obtained using the the same tuning parameters. Here we provide a high level description of the metrics. Notation and formulas for these metrics are included in Appendix D.

- *Average* R^2: Assess the model fit and linearity claim by computing the average of the explainer model R^2 values (deviance ratios from the R package *glmnet*).
- *Average Fidelity*: Measures the faithfulness of the explainer model to the complex model by comparing their predictions. Computed as the average of the explainer model fidelity metrics: a metric presented in Ribeiro et al. [23] (the weighted distance between explainer and complex model predictions for all observations in the LIME simulated data associated with an individual prediction of interest).
- *Mean Squared Explanation Error (MSEE)*: Also measures the faithfulness of the explainer model to the complex model by comparing their predictions, but only the prediction of interest is used to compute an average squared deviation between explainer and complex model predictions.

Figure 7 shows these metrics computed for each of the LIME applications to the sine data test set. The simulation methods are listed on the x-axis. The plot is faceted by metric, and the metric values are plotted on the y-axis. The colors of the points represent the rank of the simulation methods performance based on a particular metric (darker indicates a better metric value and lighter indicates a worse metric value). Higher average R^2 values are better, and lower average fidelity and MSEE values are better. This example only includes one tuning parameter: the simulation method. If more than one tuning parameter is considered, the assessment metric plot is extended by adding additional facets or levels to the x-axis.

All three metrics suggest that the kernel density performed well, but the metrics disagree for the quantile bins methods. Average R^2 and average fidelity rank the performance of the number of quantile bins the same (2 quantile bins perform the best and 6 quantile bins perform the worst). In fact, these two metrics appear to have an inverse relationship in this example. MSEE provides different ranks of the simulation methods. MSEE suggests that 6 quantile bins perform the best, and 2 quantile bins perform the worst. This is the opposite of average R^2 and average fidelity. Since average fidelity and MSEE are similar metrics, it would be expected that they would agree, but that is so in this example. This may be due to the MSEE only taking into account the prediction of interest and not the full simulated dataset. The contradiction between metrics makes it difficult to identify which simulation method explanations to trust.

Recall that Figure 6 indicated that the LIME kernel density method selected the same feature across all cases in the test set for both the first and second features. It appears that a global trend may be the best explanation for this example, which may be reasonable considering that we know that both x_1 and x_2 are

the two features that should be the features used by the random forest to distinguish between response categories.

4 | APPLICATION TO BULLET MATCHING DATA

In this section, we provide a discussion of the application of the visual diagnostics for LIME explanations to a practical data problem investigating the similarity of marks on fired bullets.

4.1 | Bullet Matching Data

In current practice, forensic firearm examiners evaluate whether two bullets are from the same source (fired from the same gun) or from different sources based on microscopic comparison of the striation patterns engraved on bullets during the firing process (see Figure 8). The process is based on a visual and therefore subjective assessment of the evidence. The lack of objective evaluation and the associated absence of established error rates has first been criticized by the National Research Council [3] and later by the President's Council of Advisors on Science and Technology [22].

In response, Hare et al. [13] proposed an automated machine learning method for bullet matching to complement a visual inspection by firearm examiners. Based on high-resolution topological scans of land engraved areas Hare et al. [13] obtain signatures of striations from two bullet lands (Figure 8). Nine features quantifying the similarity of signatures, such as the cross-correlation function, the distance between signatures, and the number of matching striae, are extracted and used to train a random forest model (available in the *bulletxtrctr* R package) to determine the probability of a comparison resulting from the same source (matching signatures) or from different sources (non-matching signatures). The model was trained on a set of scans of bullets from the James Hamby Consecutively Rifled Ruger Barrel Study [12], which includes 83028 land-to-land comparisons.

FIGURE 8 (Top left) Traditionally rifled gun barrel. The grooves and lands alternate to give bullets a spin during the firing process, which create markings (striations) on a bullet when fired. (Top right) Image of a fired bullet. The vertical stripes along the lower half of the bullet show groove and land engraved areas. The land engraved areas contain the microscopic striations created when the bullet passed through the barrel of the gun. (Bottom) Close up of a land engraved area showing striations (vertical lines).

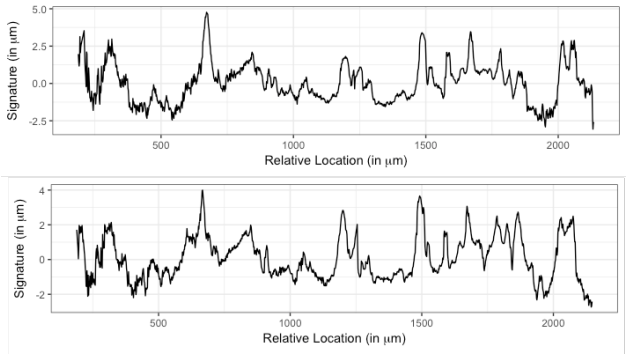

FIGURE 9 Example bullet signatures. The bullet signatures are from the same barrel-land and therefore have very similar patterns. The Hare et al. [13] random forest uses features that measure the similarity between two such signatures.

4.2 | Application of LIME to Bullet Matching Data

Since firearm identification is commonly used as evidence for convictions in court cases, it is important to be able to understand and assess a model used to quantify the probability that a bullet is fired from a gun. LIME explanations would provide a local explanation for an individual prediction, but just as it is important to assess the model for this high-stakes application, it is also important to assess the LIME explanations. We will demonstrate an assessment of LIME explanations using the visual diagnostics introduced in this paper.

We apply the random forest model from Hare et al. [13] to another set of bullets from the Hamby study with 432 rows of land comparisons. We first consider a global visualization of the relationship between the random forest predictions and the model features by creating a parallel coordinate plot of the training data (top facet) and testing data (bottom facets) predictions (Figure 10). The majority of same source observations with random forest probabilities close to 1 have a clear pattern of corresponding feature values. The known same source observations where the random forest is wrong (does not return

FIGURE 10 Parallel coordinate plots of the Hare et al. [13] random forest predictions from the training (top facet) and testing (bottom facet) data. The observations are separated by known matching (right column) and non-matching (left column) signatures. The y-axis shows the standardized feature values, and the x-axis shows the features used to fit the random forest (ordered by random forest impurity based feature importance). Each line corresponds to an observation, and the line color represents the associated random forest probability. There are clear relationships between the feature values and the random forest probabilities.

probabilities close to 1) have feature values that reflect those of observations where same source is known to be false.

LIME is implemented using the R package *limeaid* to apply LIME multiple times to all test set observations using different tuning parameters: 12 sampling methods (2-6 equally spaced bins, 2-6 quantile bins, kernel density estimation, and normal approximation) and 3 Gower exponents (0.5, 1, and 10). Thus, a total of $12 \times 3 = 36$ different applications of LIME were performed. We specify that each LIME explanation return 3 features and feature selection is performed using the highest weights method (default option).

4.3 | LIME Assessment Visualizations

To get an overview of the LIME explanations from the 36 applications, we consider a feature heatmap (Figure 11). In addition to facets for simulation method and order of feature selected by LIME, this plot includes a vertical facet for Gower power and a horizontal facet for whether the observation is a known match or non-match. This plot highlights several key features of the LIME explanations from the bullet matching dataset.

First, the density simulation methods produce the same explanations for almost all cases and LIME tuning parameters suggesting the LIME explanations are global and not local. Second, within a bin based simulation method, the features selected by LIME for an observation often vary by the number of bins but do not appear to vary by the Gower power. With the equal bins, there are vertical stripes that suggest a dependence of the LIME explanations on the number of bins. The vertical stripes are not as apparent with the quantile bins. Lastly, there are clear differences between the LIME explanations produced by the bin based simulation methods for the matches and non-matches. This suggests that the features the random forest uses to classify a match or non-match are different.

To try to identify a set of LIME tuning parameters with the most trustworthy set of explanations, an assessment metric plot is considered (Figure 12). All density simulation methods performed well based on the three metrics. This is an interesting result, since Figure 11 shows that the density methods results in global explanations. The metric values for the bin based methods do not agree across metrics. For example, 2 and 3 quantile bins performed well according to the average R^2 but poorly according to MSEE. As a result, with the bin based methods, it is difficult to know which method to recommend. The applications using a power of 0.5 perform the best or as well as the other powers across all simulation methods suggesting that the power that leads to a more global explanation is preferred by LIME.

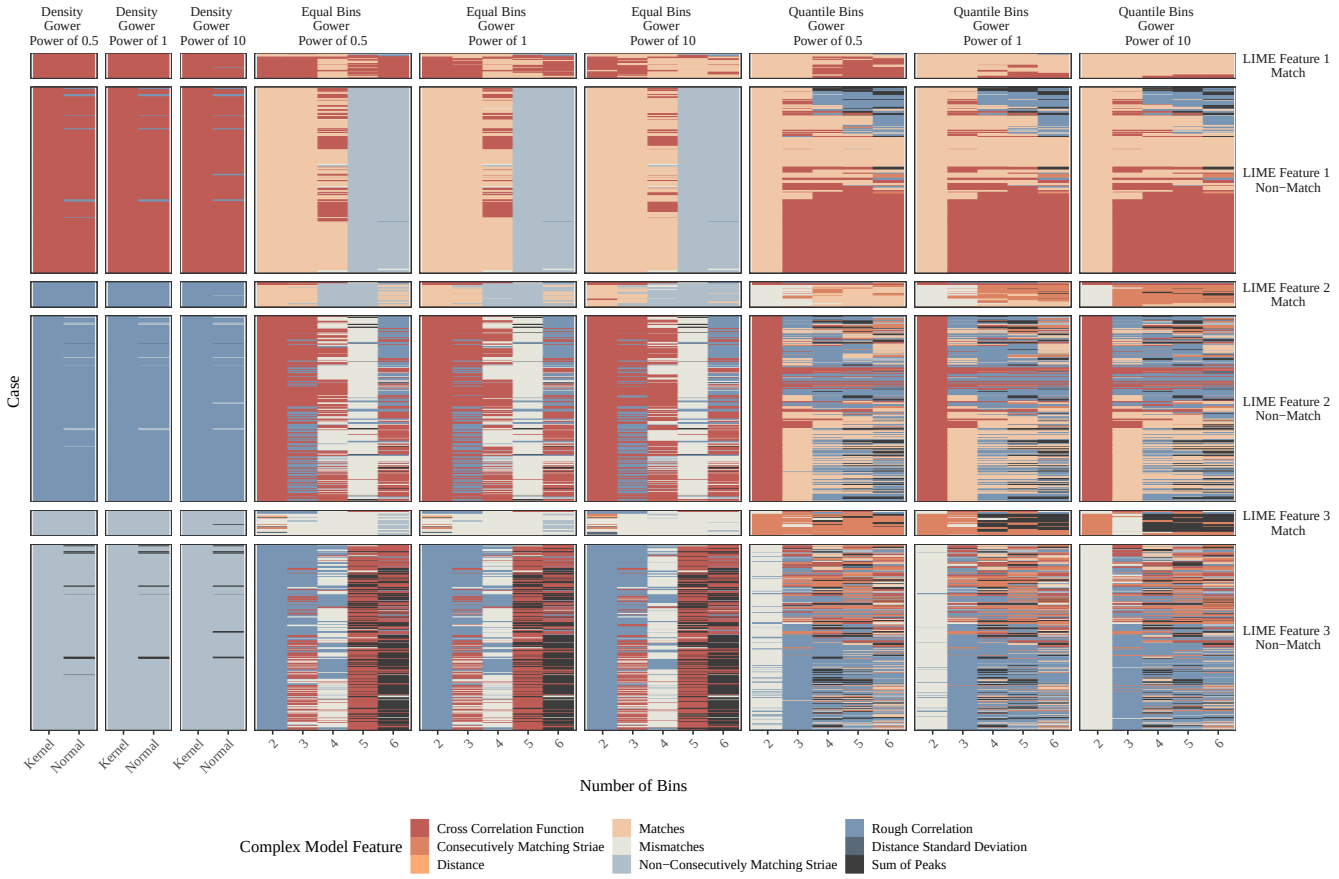

FIGURE 11 Feature heatmap of the 36 LIME applications to the bullet comparison data test set. In addition to faceting the results by simulation method and LIME feature selection order, facets for the Gower power and whether the observation is a match or non-match are included. The vertical stripes of features selected indicate a dependence between the LIME explanations and tuning parameters.

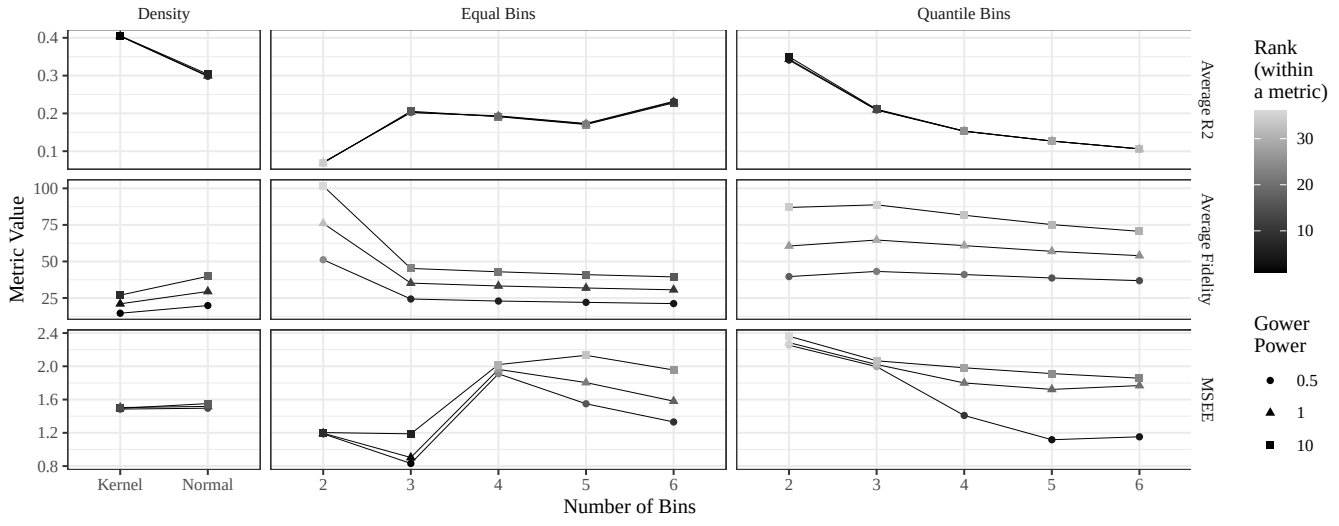

FIGURE 12 Plot of LIME assessment metrics for the applications of LIME on the bullet comparison data test set. The density simulation methods perform well for all metrics. The bin based metrics often do not agree in terms of performance across metrics.

Perhaps we can say that the kernel density method, 3 equal bins, or 3 quantile bins are some of the better performing methods, but unfortunately, Figure 12 leaves us without a clear indication of which set of explanations to use to explain the random forest. To better understand the explanations provided by these three methods, we take a closer look at explanations from these methods for two observations of interest, which will be referred to as case M (a known match) and case NM (a known non-match).

Figure 13 include the explanation plots (from *lime*) and explanation scatterplots for cases M and NM. Plots for 3 equal bins and 3 quantile bins are included. Plots for the kernel density simulation method are included in Figure B2 . The scatterplots provide several insights into the LIME explanations. Here we focus on the explanation for case NM with 3 quantile bins, but similar statements can be made about the other plots.

The explanation plot from *lime* for case NM with 3 quantile bins shows that an observed CCF value greater than 0.320 supports a random forest prediction in favor of a match, and the scatter plots show that many of the simulated values with CCF greater then 0.320 have random forest predictions greater than 0.5. Additionally, the LIME explanation indicates that a value of matches less than or equal to 1.66 contradicts a prediction in favor of a match, and the scatter plots show that almost all simulated values with a value of matches less than or equal to 1.66 have random forest predictions close to 0. While the LIME explanation makes sense based on the observations made from the explanation scatter plot, the plot also indicates that LIME falls short of providing a good explanation of the random forest prediction for case NM. Based on the scatter plot of CCF versus matches, a better explanation would be that because case NM has a value of CCF less than 0.8 and a value of matches less than 7, the random forest provides a prediction supporting a non-match. The relationship between CCF and rough correlation does not provide much evidence to support the random forest prediction one way or the other, but there is a pentagon shaped region at the bottom of the scatter plot of matches versus rough correlation with mostly predictions close to 0 that case NM falls and supports the random forest prediction of a non-match.

Without applying LIME with multiple tuning parameters to the bullet test data or viewing diagnostic plots of the LIME explanations, it may be very possible to formulate reasons why the LIME explanations make sense. However, the sequence of plots in this section (Figures 11 , 12 , and 13) suggest that we should be cautious to trust any of these LIME explanations. It appears that either LIME needs to be further tuned to provide trustworthy and good explanations, or a different approach may provide better insight.

5 | DISCUSSION

This paper highlights that while an explainer model is meant to provide clarity, it actually adds another layer of complexity to predictive models by requiring yet another model that needs to be assessed. Without an assessment of the explainer model, LIME is a black-box procedure of its own requiring blind trust in the explainer model. We suggest the use of visual diagnostics to counteract the black-box nature of LIME and provide three diagnostic plots.

The visualizations are intended to provide insight on how LIME works, assess the ability of the explainer model to capture the complex predictive model, and compare LIME explanations produced by different tuning parameters. While the visualizations accomplish these tasks, they also expose examples of the failings of LIME. To address the discovered failings of LIME, we will reconsider each of the claims about the performance of LIME made by Ribeiro et al. [23] in light of the insights gained from the diagnostic visualizations.

As previously discussed, the **interpretability** of the LIME explanations can be controlled by the complexity of the explainer model. For example, the number of bins selected for simulation can control the interpretability of the explanations. If too many bins are selected, the bin range that is reported in the LIME explanation will be too small to be meaningful in the context of the feature. An appropriate choice of the number of bins will keep the bin range meaningful. Thus, the claim of interpretability does not need to be assessed using the visualizations. However, diagnostic visualizations do present a different perspective on the meaning of interpretability.

Even though an explanation will be interpretable as long as the complexity of the explainer model is appropriately chosen, a lack of understanding of the process used to create the explanation could lead to an incorrect interpretation of the explanation. For example, the visualization of a LIME explanation available from the *lime* R package [21] (shown in Figures 4 and 13) is a major simplification of the explainer model, which could lead to under-interpreted or misinterpreted LIME explanation. Supplementing Pedersen and Benesty [21]'s compact visualization of the explanation with an explanation scatterplot that shows a more detailed visualization of the explainer model (such as Figures 4 and 13) promotes a full interpretation of the explanation.

Even with an explainer model that is interpreted correctly, the interpretation is worthless if the explainer model is not **faithful** to the complex model. This claim can be assessed using the diagnostic plots suggested in this paper. Many of the visualizations in this paper highlight problems with the faithfulness of the explainer models. The explanation scatterplots allow for a comparison of the explainer model to the complex model. The examples in this paper show cases where

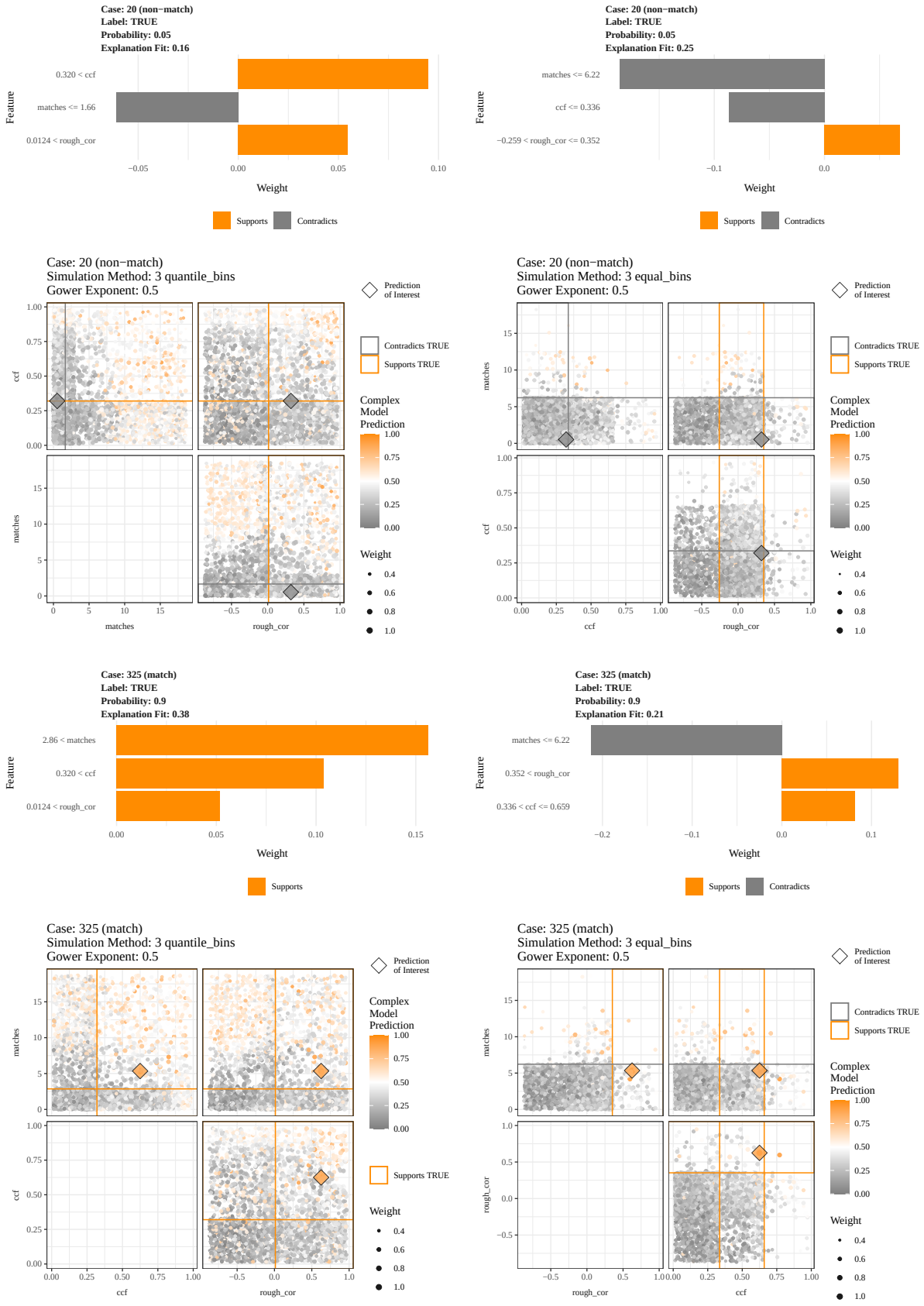

FIGURE 13 Plots of LIME explanations and explanation scatterplots for one case M and NM in the bullet test data. The first column contains plots associated with 3 quantile bins and the second with 3 equally spaced bins. The plots provide insights into the LIME explanations and allow for the assessment of the explanation quality.

the explainer model bins do not accurately capture the classification boundaries near the predictions of interest of the random forest models by oversimplifying the model (Figures 4 and 13). Using less bins would clearly not help improve the faithfulness of the explainer model in these examples, and while an increase in bins would lead to a finer resolution of the random forest classification boundaries, interpretability of the explainer model would quickly be lost. Perhaps this could be improved by allowing the bin creation to account for the relationships between the features and response variable or a different number of bins for each feature.

In addition to assessing faithfulness by visually comparing the complex and explainer model, we propose a visual comparison of two faithfulness metrics (MSEE and average fidelity). The examples of faithfulness metric comparisons in this paper (Figures 7 and 12) both produced confusing results. The density based simulation methods resulted in the best or close to the best performance even though the feature heatmaps (Figures 6 and 11) showed that the density simulation methods produced global explanations with no variation in features selected by LIME. For the bin based simulation methods, the two metrics often did not agree or contradicted one another, which makes it difficult to decide on a recommendation of a set of tuning parameters that produces the explanations with the most faithful explainer model.

The metric comparison plot also includes a comparison of average R^2 values, which is a metric that can be used to assess the claim of **linearity**. Most of the average R^2 values in the examples from this paper are below 0.5 suggesting a poor linear fit of the explainer models. The poor linear fit of the explainer model is also seen with the residual plot (Figure C3).

The final claim, **localness**, is addressed by the feature heatmap and metric comparison plot. As stated, the feature heatmap revealed that the density simulation methods in the examples of this paper resulted in global explanations where the same features were repeatedly chosen across all (or almost all) observations in the set of explanations. This finding agrees with that of [16] who found LIME produced global explanations with the normal approximation simulation method. For the bin based simulation methods in the bullet data example, the feature heatmap showed that the features chosen for the explanations varied between the two classification categories (match versus non-match). This is an interesting finding that suggests that different features can play a role in the predictions of observations in different response categories, but the patterns of features selected by LIME within a classification category do not vary. Again, this is a suggestion of global explanations. Furthermore, while the metric comparison plot in the bullet example did not provide agreement between metrics on a best bin based method, all metrics agree that a Gower power of 0.5 for computing the model weights associated with distance of a simulated data point from the prediction of interest is best. This suggests that a less local explanation provided a better explanation of the performance of the random forest.

Some of the visualizations in the paper generalize easily to any application of LIME such as the feature heatmap and metric plot. Other plots such as the visualizations of the LIME procedure would require extensions such as the use of scatterplot matrices to compare explanations with more than two features. The addition of interactivity to the diagnostic plots would provide additional enhancement of the assessment process. For example, a diagnostic plot that provides a summary of multiple LIME explanations, such as the feature heatmap, could be displayed and clicked on to reveal more detailed figures associated with individual predictions of interest, such as plots of the simulated data and explainer model.

The largest limitation to the diagnostic visualizations is the dimensionality of the data shown, both in the number of dimensions or features as well as the number of observations. Fortunately, in the situation of LIME, both of these aspects are rather well controlled: LIME relies heavily on simulations to generate data scenarios that are close to the data observed but exhibits variability. Effects from overplotting should be relatively mild, because output from simulations is shown, which is expected to be (relatively) continuous such that overplotting only occurs for points with (relatively) similar values. What do you mean by continuous here?. XXX Does that extra explanation help? In that respect, the diagnostics shown for the sine data and in the bullet example are representative of what is expected. But in cases where overplotting does become problematic, the user could either simply reduce the size of the simulations or use some well-studied binning techniques in the visualizations, as discussed for example in Carr et al. [2] or Unwin et al. [28].

While it would be ideal if LIME could be used as a method to provide easily understandable explanations for black-box models as [23] claim, that dream is not yet a reality. The examples using diagnostic plots to assess LIME in this paper show frequent issues with LIME. We hope that our plots provide motivation to assess LIME explanations, to not blindly use the default settings (even if it is not clear which tuning parameters to use), and to encourage work on improving LIME, so that it can be a lime and not a lemon.

ACKNOWLEDGMENTS

HH was partially funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) through Cooperative Agreement 70NANB20H019 between NIST and Iowa State University, which includes activities carried out at Carnegie

Mellon University, Duke University, University of California Irvine, University of Virginia, West Virginia University, University of Pennsylvania, Swarthmore College and University of Nebraska, Lincoln.

Author contributions

A taxonomy of author contributions is suggested by Elsevier - I put the link in an email

Katherine Goode, Heike Hofmann: Conceptualization, Methodology, Investigation, Writing- Reviewing and Editing. **Katherine Goode**: Writing Initial Draft, Software.

Financial disclosure

None reported.

Conflict of interest

The authors declare no potential conflict of interests.

SUPPORTING INFORMATION

The following supporting information is available as part of the online article:

References

[1] Alvarez-Melis, D. and T. S. Jaakkola, 2018: On the robustness of interpretability methods. [15].
URL https://arxiv.org/abs/1806.08049

[2] Carr, D. B., R. J. Littlefield, W. L. Nicholson, and J. S. Littlefield, 1987: Scatterplot Matrix Techniques for Large N. *Journal of the American Statistical Association*, **82**, no. 398, 424–436, doi:10.1080/01621459.1987.10478445.

[3] Committee on Identifying the Needs of the Forensic Sciences, National Research Council, 2009: *Strengthening Forensic Science in the United States: A Path Forward*. https://www.ncjrs.gov/pdffiles1/nij/grants/228091.pdf.

[4] Emerson, J. W., W. A. Green, B. Schloerke, J. Crowley, D. Cook, H. Hofmann, and H. Wickham, 2013: The Generalized Pairs Plot. *Journal of Computational and Graphical Statistics*, **22**, no. 1, 79–91, doi:10.1080/10618600.2012.694762.

[5] Fisher, A., C. Rudin, and F. Dominici, 2019: All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, **20**, no. 177, 1–81.
URL http://jmlr.org/papers/v20/18-760.html

[6] Friedman, J. H., 2001: Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, **29**, no. 5, 1189–1232, doi:10.1214/aos/1013203451.

[7] Gilpin, L. H., D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, 2018: Explaining Explanations: An Overview of Interpretability of Machine Learning. *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, F. Bonchi and F. Provost, Eds., IEEE, 80–89.

[8] Goode, K., 2020: *limeaid: Diagnose LIME Explanations*. R package version 0.0.1.

[9] Goodman, B. and S. Flaxman, 2017: European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, **38**, no. 3, 50–57, doi:10.1609/aimag.v38i3.2741.
URL https://www.aaai.org/ojs/index.php/aimagazine/article/view/2741

[10] Gower, J. C., 1971: A general coefficient of similarity and some of its properties. *Biometrics*, **27**, 857–871, doi:10.2307/2528823.
URL https://www.jstor.org/stable/2528823

[11] Guidotti, R., A. Monreale, S. Ruggieri, F. Turini, D. Pedreschi, and F. Giannotti, 2018: A survey of methods for explaining black box models. *ACM Computing Surveys*, **51**, no. 5, doi:10.1145/3236009.

[12] Hamby, J. E., D. J. Brundage, and J. W. Thorpe, 2009: The identification of bullets fired from 10 consecutively rifled 9mm ruger pistol barrels: A research project involving 507 participants from 20 countries. *AFTE Journal*, **41**, no. 2, 99–110.

[13] Hare, E., H. Hofmann, and A. Carriquiry, 2017: Automatic matching of bullet land impressions. *Annals of Applied Statistics*, **11**, no. 4, 2332–2356, doi:10.1214/17-AOAS1080.

[14] Jensen, M. S., R. Yao, W. N. Street, and D. J. Simons, 2011: Change blindness and inattentional blindness. *Wiley interdisciplinary reviews. Cognitive science*, **2**, no. 5, 529–46, doi:10.1002/wcs.130.

[15] Kim, B., K. R. Varshney, and A. Weller, Eds., 2018: *2018 ICML Workshop on Human Interpretability in Machine Learning (WHI 2018)*, ICML.

[16] Laugel, T., X. Renard, M. Lesot, C. Marsala, and M. Detyniecki, 2018: Defining locality for surrogates in post-hoc interpretablity. [15].
URL http://arxiv.org/abs/1806.07498

[17] Liaw, A. and M. Wiener, 2002: Classification and regression by randomforest. *R News*, **2**, no. 3, 18–22.
URL https://CRAN.R-project.org/doc/Rnews/

[18] Ming, Y., 2017: *A Survey on Visualization for Explainable Classifiers*. Ph.D. thesis.

[19] Mohseni, S., N. Zarei, and E. D. Ragan, 2018: A survey of evaluation methods and measures for interpretable machine learning.

[20] Molnar, C., 2019: *Interpretable Machine Learning*.

[21] Pedersen, T. L. and M. Benesty, 2020: *lime: Local Interpretable Model-Agnostic Explanations*. Https://lime.data-imaginist.com, https://github.com/thomasp85/lime.

[22] President's Council of Advisors on Science and Technology, 2016: *Report on forensic science in criminal courts: Ensuring scientific validity of feature-comparison methods*. https://obamawhitehouse.archives.gov/sites/default/files/microsites/ostp/PCAST/pcast_forensic_science_report_final.pdf.

[23] Ribeiro, M. T., S. Singh, and C. Guestrin, 2016: "why should I trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 1135–1144.

[24] Sapra, R., 2014: Using r2 with caution. *Current Medicine Research and Practice*, **4**, no. 3, 130–134, doi:10.1016/j.cmrp.2014.06.002.

[25] Simon, N., J. Friedman, T. Hastie, and R. Tibshirani, 2011: Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, **39**, no. 5, 1–13.
URL http://www.jstatsoft.org/v39/i05/

[26] Simonyan, K., A. Vedaldi, and A. Zisserman, 2013: Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps.

[27] Sweller, J., 2011: Chapter two - cognitive load theory. Academic Press, volume 55 of *Psychology of Learning and Motivation*, 37 – 76.
URL http://www.sciencedirect.com/science/article/pii/B9780123876911000028

[28] Unwin, A. R., M. Theus, and H. Hofmann, 2006: *Graphics of Large Datasets: Visualizing a Million*. Springer, New York.

[29] Welling, S. H., H. H. F. Refsgaard, P. B. Brockhoff, and L. H. Clemmensen, 2016: Forest Floor Visualizations of Random Forests.

[30] Štrumbelj, E. and I. Kononenko, 2014: Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, **41**, no. 3, 647–665, doi:10.1007/s10115-013-0679-x.

How to cite this article: Goode K., H. Hofmann, 2019, Visual Diagnostics of a Model Explainer – Tools for the Assessment of LIME Explanations, *Stat Anal Data Min: The ASA Data Sci Journal*, *volume, number and page*.

APPENDIX

A LIME TUNING PARAMETER OPTIONS

The following tuning parameters for the LIME algorithm are available in the *lime* R package.

- Data simulation methods:
  - Equally spaced bins: observations are uniformly sampled from equally spaced bins (number of bins may be specified)
  - Quantile bins: observations are uniformly sampled from quantile bins (number of bins may be specified)
  - Normal density approximation: observations are sampled from a normal distribution with mean and standard deviation computed from the corresponding feature
  - Kernel density approximation: observations are sampled from an kernel density approximation of the corresponding feature
- Number of observations to simulate
- Distance metric for determining proximity to the prediction of interest: Gower distance (where the power may be specified) or exponential kernel (where the kernel width may be specified)
- Number of features to return in an explanation
- Feature selection method for determining the features to return in an explanation: forward selection applied to a ridge regression, highest weights in ridge regression, LASSO, classification/regression tree splits

B EXPLANATION SCATTERPLOTS UNDER OTHER SIMULATION SCENARIOS

Section 3.2 introduces explanation scatterplots under the default simulation method in the *lime* R package: four quantile bins. The structure of an explanation scatterplot remains the same for if any bin based simulation method is used to create the simulated data (any number of quantile or equally spaced bins). However, if the kernel density or normal approximation simulation methods are used as the simulation method, the format of the explanation scatterplot changes. In the density based simulation method scenarios, LIME uses the standardized versions of the predictor variables to fit the explainer model. Thus, the explainer model needs to be represented differently in the explanation scatterplot.

When the kernel density or normal approximation simulation methods are applied, the explanation scatterplot depicts the complex model by plotting the complex model predictions versus a feature selected in LIME the explanation from the simulated data. The explainer model is included as a line on the figure where all features excluding the one plotted on the x-axis are set to the observed values of the prediction of interest. An explanation scatterplot is created for each feature included in the LIME explanation. As with the bin based simulation method, the size of the points represent the weight assigned by LIME.

Figure B1 provides example explanation scatterplots for each feature in the default LIME explanation obtained using the kernel density simulation method for the sine data prediction of interest.

Figure B2 includes explanation scatterplots for the explanations generated using kernel density simulation for the bullet example cases M and NM discussed in Section 4.3.

C EXPLAINER MODEL RESIDUAL PLOT

In order to assess the claim of linearity for the sine data prediction of interest discussed in Section 3.2, we use one of the most basic diagnostics in a statistician's tool box and draw a residual plot for the explainer model. This is shown in Figure C3 with the explainer model residuals on the y-axis and explainer model predictions on the x-axis. The points along the x-axes have been jittered to ease the effect of the over-plotted points in the visualization. There is a clear increasing trend in the residuals as the explainer model predictions increase. This is a clear violation of the linearity assumption with the ridge regression model.

D DETAILS ON ASSESSMENT METRICS

Suppose f is a complex model, and let $\mathbf{X}$ be a matrix of observed data with K features and E observations where x_e is an observed feature vector for observation e. Let $f(x_e)$ be the complex model prediction for observation e. It is of interest to explain the predictions made by f applied to X using LIME.

For x_e and set of tuning parameters t, let $\mathbf{X}'_{e,t}$ be the LIME simulated dataset with K features and S observations such that $x'_{e,t,s}$ is the feature vector for simulated data observation s.

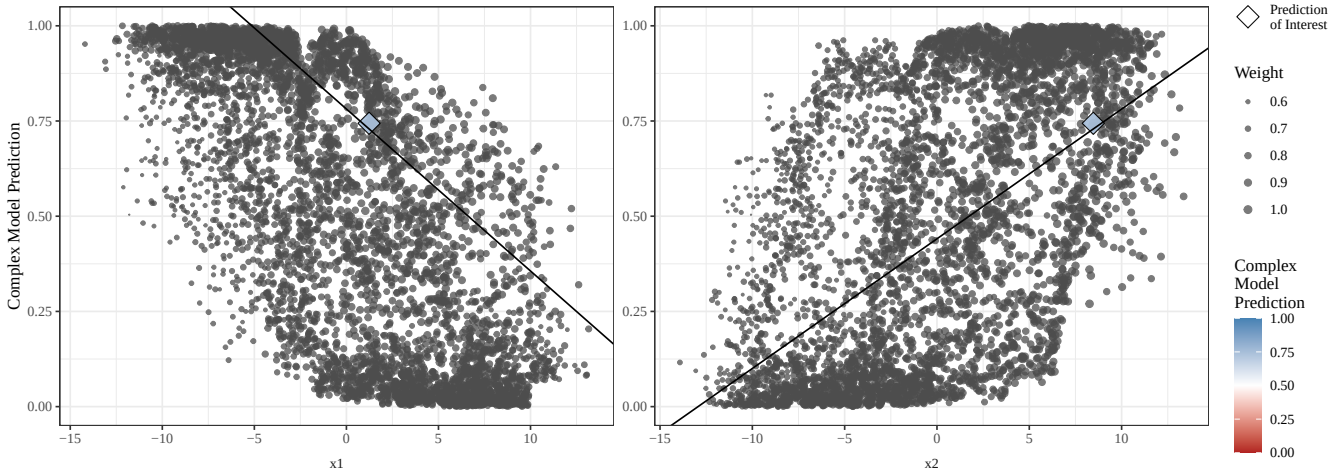

FIGURE B1 Explanation scatterplots for the sine data prediction of interest with the kernel density simulation method.

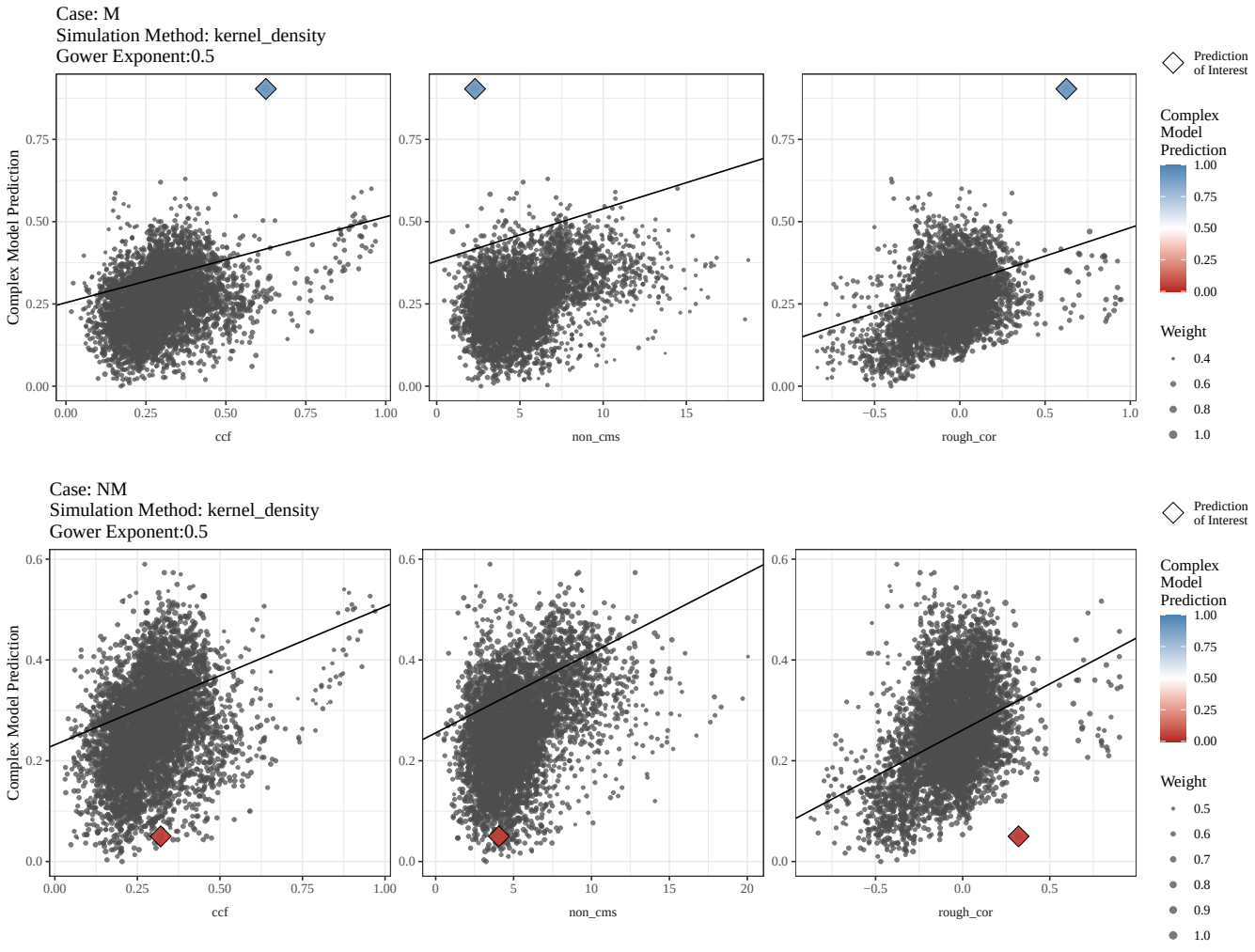

FIGURE B2 Explanation scatterplots for LIME explanations using kernel density simulation for the cases M and NM of the bullet comparison.

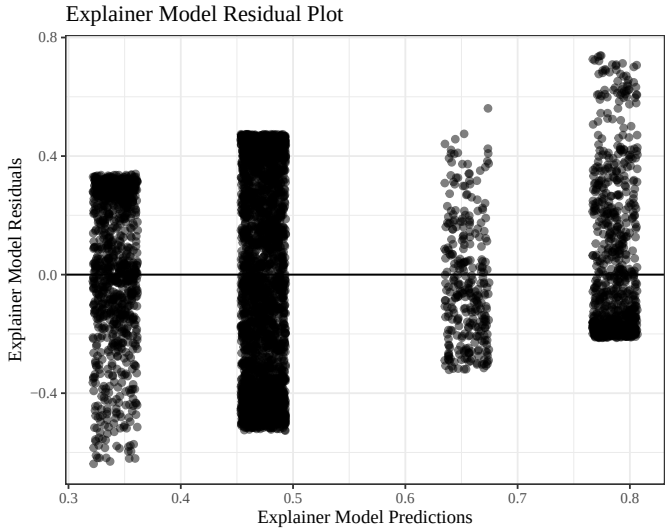

FIGURE C3 Residual plot of the explainer model associated with sine data prediction of interest from Section 3.2. The residuals are plotted against the predicted values. The points are jittered in the x-direction to alleviate the overplotting of points. There is an upward trend in the residuals as the explainer model predictions increase, which suggests a violation of the linearity assumption.

Let $\mathbf{Z}'_{e,t}$ be the matrix of interpretability transformed simulated data with K features and S observations such that $z'_{e,t,s}$ is the interpretability transformed feature vector for observation/explanation e, tuning parameter t, and simulated data observation s. Note that $z_{e,t}$ will represent the interpretability transformed version of x_e.

Next, let ω_t represent a proximity distance metric. Then $\omega_t\left(x_e, x'_{e,t,s}\right)$ is the weight assigned to $x'_{e,t,s}$, which is the proximity between x_e and $x'_{e,t,s}$. Allow $g_{e,t}$ to be the explainer model for an explanation e and tuning parameter t. Thus, $g_{e,t}\left(z'_{e,s,t}\right)$ is the explainer model prediction for the interpretability transformed simulated data observation s.

For a set of E explanations and a set of tuning parameters t, we define the assessment metrics as follows:

Average R^2 is denoted as R^2_{ave} and computed as

$$R^2_{\text{ave}} = \frac{1}{E}\sum_{e=1}^{E} R^2_{e,t}$$

where $R^2_{e,t}$ is the R^2 value for $g_{e,t}$.

Average fidelity is denoted by $\mathcal{L}_{\text{ave}}$ and computed as

$$\begin{aligned}\mathcal{L}_{\text{ave}} &= \frac{1}{E}\sum_{e=1}^{E}\mathcal{L}(f,\, g_{e,t},\, \pi_t)\\ &= \frac{1}{E}\sum_{e=1}^{E}\sum_{s=1}^{S}\omega_t\left(x_e, x'_{e,t,s}\right)\left(f\left(x'_{e,t,s}\right) - g_{e,t}\left(z'_{e,t,s}\right)\right)^2.\end{aligned}$$

Mean squared explanation error is denoted by MSEE and computed as

$$MSEE = \frac{1}{E}\sum_{e=1}^{E}\left(f\left(x_e\right) - g_{e,t}\left(z_{e,t}\right)\right)^2.$$