Model Agnostic Supervised Local Explanations

Gregory Plumb
CMU
gdplumb@andrew.cmu.edu

Denali Molitor
UCLA
dmolitor@math.ucla.edu

Ameet Talwalkar
CMU
talwalkar@cmu.edu

Abstract

Model interpretability is an increasingly important component of practical machine learning. Some of the most common forms of interpretability systems are example-based, local, and global explanations. One of the main challenges in interpretability is designing explanation systems that can capture aspects of each of these explanation types, in order to develop a more thorough understanding of the model. We address this challenge in a novel model called MAPLE that uses local linear modeling techniques along with a dual interpretation of random forests (both as a supervised neighborhood approach and as a feature selection method). MAPLE has two fundamental advantages over existing interpretability systems. First, while it is effective as a black-box explanation system, MAPLE itself is a highly accurate predictive model that provides faithful self explanations, and thus sidesteps the typical accuracy-interpretability trade-off. Specifically, we demonstrate, on several UCI datasets, that MAPLE is at least as accurate as random forests and that it produces more faithful local explanations than LIME, a popular interpretability system. Second, MAPLE provides both example-based and local explanations and can detect global patterns, which allows it to diagnose limitations in its local explanations.

1 Introduction

Leading machine learning models are typically opaque and difficult to interpret, yet they are increasingly being used to make critical decisions: e.g., a doctor's diagnosis (life or death), a biologist's experimental design (time and money), or a lender's loan decision (legal consequences). As a result, there is a pressing need to understand these models to ensure that they are correct, fair, unbiased, and/or ethical. Although there is no precise definition of interpretability and user requirements are generally application-specific, three of the most common types of model explanations are:

1. **Example-based**. In the context of an individual prediction, it is natural to ask: *Which points in the training set most closely resemble a test point or influenced the prediction?* Nearest neighbors and influence function based methods are archetypal methods that naturally lead to example-based explanations [1, 2, 12].

2. **Local**. Alternatively, we may aim to understand an individual prediction by asking: *If the input is changed slightly, how does the model's prediction change?* Local explanations are typically derived from a model directly (e.g., sparse linear models), or from a local model that approximates the predictive model well in a neighborhood around a specific point [1, 17].

3. **Global**. To gain an understanding of a model's overall behavior we can ask: *What are the patterns underlying the model's behavior?* Global explanations usually take the form of a series of rules [13, 18].

Example-based explanations are clearly distinct from the other two explanation types, as the former relies on sample data points and the latter two on features. Further, local and global explanations
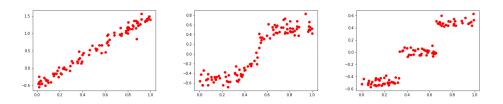
Figure 1: Toy datasets. (a) Linear. (b) Shifted Inverse Logistic (SIL). (c) Step Function.

themselves capture fundamentally different characteristics of the predictive model. To see this, consider the toy datasets in Fig. 1 generated from three univariate functions.

Generally, local explanations are better suited for modeling smooth continuous effects (Fig. 1a). For discontinuous effects (Fig. 1c) or effects that are very strong in a small region (Fig. 1b), which can be approximated well by discontinuities, they either fail to detect the effect or make unusual predictions, depending on how the local neighborhood is defined. We will call such effects *global patterns* because they are difficult to detect or model with local explanations. Conversely, global explanations are better suited for global patterns because these discontinuities create natural 'rules,' and are less effective at explaining continuous effects because explanation rules must introduce arbitrary feature discretization/binning. Most real datasets have both continuous and discontinuous effects and, therefore, it is crucial to devise explanation systems that can capture, or are at least aware of, both types of effects.

In this work, we propose a novel model explanation system that draws on ideas from example-based, local, and global explanations. Our method, called MAPLE[1], is a *supervised neighborhood* approach that combines ideas from local linear models and ensembles of decision trees. One of the distinguishing features of MAPLE is that, for a particular prediction, it assigns weights to each training point; these weights induce a probability distribution over the input space that we call the *local training distribution*. MAPLE is endowed with several favorable properties described below:

- It avoids the typical trade-off between model accuracy and model interpretability [8, 15, 16], as it is a highly accurate predictive model (on par with leading tree ensembles) that simultaneously provides example-based and local explanations.

- While it does not provide global explanations, it does detect global patterns by leveraging its local training distributions, thus distinguishing it from other local explanation methods. As a consequence, it can *diagnose* limitations of its local explanations in the presence of global patterns. Moreover, it offers global feature selection, which can help detect data leakage [10] or possible bias [18].

- In some settings, we want to represent a predictive model with a small number of exemplar explanations [17] and, when asked to explain a new test point, must decide which of these exemplar explanations to use. Its local training distribution allows us to make that decision in a principled fashion, thereby addressing another key weakness of existing local explanation systems [18].

- In addition to providing faithful self-explanations, it can also be deployed effectively as a black-box explainer by simply training it on the predictions of a black-box predictive model instead of the actual labels. We find that it produces more faithful explanations than LIME [17], a commonly used model-agnostic algorithm for generating local explanations.

2 Background and Related Work

We divide the background material into two sections. The first outlines interpretability and the second reviews the random forest literature that is most relevant to MAPLE.

---

[1]The source code for our model and experiments is at https://github.com/GDPlumb/MAPLE.

2.1 Interpretability

We say that a function is **interpretable** if it is human simulatable [15]. This definition requires that a person can 1) carry out all of the model calculations in reasonable time, which rules out functions that are too complicated, and 2) provide a semantic description of each calculation, which rules out using features that do not have well understood meanings. Generally, sparse-linear models, small decision trees, nearest neighbors, and short rule lists are all considered to be human simulatable.

Based on the ideas in [15, 17], we define a **local explanation** at x, denoted $exp_x()$, as an interpretable function that approximates $pred()$ well for x' in a neighborhood of x where $pred()$ is the predictive model being explained. There are two main challenges when using this type of explanation. The first is accurately modeling or detecting global patterns. Intuitively, local explanations based on supervised neighborhood methods will fail to detect global patterns and those based on unsupervised neighborhoods will be inaccurate near them because their neighborhoods are, respectively, either partitioned on the discontinuity or contain points on both sides of it. More details about and a visualization of this are in Sec. 5.4. The second challenge is determining if an explanation generated at one point can be applied at a new point [18]. We demonstrate how MAPLE addresses these challenges in Sec. 4.

The most closely related work to ours is LIME [17], which provides a local explanation by fitting a sparse linear model to the predictive model's response via sampling randomly around the point being explained. The authors of [17] also demonstrate how to use local explanations in a variety of practical applications. They also define SP-LIME which summarizes a model by finding a set of points whose explanations (generated by LIME) are diverse in their selected features and their dependence on those features. However, this does not address the difficulties local explanations have with global patterns because the individual explanations being chosen all are unaware of them.

Fundamentally, the problem of identifying a good local explanation is a causal question, which is typically very difficult to answer since most models are not causal. However, the local explanation is not trying to find causal structure in the data, but in the model's response. This makes the problem feasible because we can freely manipulate the input and observe how the model's response changes. However, most explanation systems are not evaluated in a way that is consistent with this goal and use the standard evaluation metric: $\mathbb{E}_x[loss(exp_x(x), pred(x))]$. To address this issue, we define the **causal local explanation metric** as

$$\mathbb{E}_{x,x' \sim p_x}[loss(exp_x(x'), pred(x'))]. \tag{1}$$

This metric is based on sampling x' from p_x, which is a distribution centered around x, and encourages the explanation generated at x to accurately predict the model's value at x'. Interestingly, performing well on the standard metric does not guarantee doing well on the casual metric. To see this, consider a local linear explanation with the form $exp_x(x') = 0^T x' + pred(x)$. While it performs perfectly on the standard evaluation metric, its performance on the causal metric can be arbitrarily bad. This is because all of the active effects (the features that, if perturbed, significantly effect the model's response) are rolled into the bias term $pred(x)$ rather than being given a non-zero coefficient. This explanation does not tell us anything about why the model made its prediction and, consequently, we want a metric that does not select it.

Based on the methods in [13, 18], we define a **global explanation** as a set of rules that generally hold true for $pred()$ for all or a well defined subset of the input space. Two of the main challenges of using global explanations are 1) adequately covering the input space and 2) properly processing the data to allow the rules to be meaningful. Anchors [18], which approximates the model with a set of if-then rules, is an example of a method that provides global explanations. It has the advantage of providing easily understood explanations and it is simple to determine if an explanation applies for a specific instance. Further, [18] shows that it is possible to choose these rules such that they have very high precision (at the cost of reduced coverage). While MAPLE does not directly offer global explanations, we show how to use it to detect global patterns in Sec. 4.1, which are the patterns global explanations represent best.

Based on [1, 12, 15], we define an **example-based explanation** as any function that assigns weights to the training points based on how much influence they have on the predictive model or on an individual prediction. The most general form of an example-based explanation is influence functions [6]. Influence functions study how the model or prediction would change if a training point was up-weighted infinitesimally, but require model differentiability and convexity to work. However,

it was recently shown that these assumptions can be relaxed and that influence functions can be used for understanding model behavior, model debugging, detecting dataset errors, and creating visually indistinguishable adversarial training examples [12]. More specific strategies for methods of generating example-based explanations are Case Based Reasoning [2], K Nearest Neighbors [1], and SP-LIME [17]. Notably, MAPLE naturally provides influence functions via the weights it assigns to the training points for a particular test-point/prediction.

2.2 Random Forests: Feature Selection and Local Models

Due to their accuracy and robustness, random forests [4] have been a popular and effective method in machine learning. Unfortunately, they are not generally considered to be interpretable because they aggregate many decision trees, each of which is often quite large. However, they yield a measure of global variable importance and they can be viewed as a way of doing supervised neighborhood selection. We use both of these aspects in defining our method.

The permutation based importance measure initially proposed in [4] determines the feature importance by considering the performance of the random forest before and after a random permutation of a predictor. Another popular variant was proposed in [9] which works by summing the impurity reductions over each node in the tree where a split was made on that variable, while adjusting for the number of points in the node, and then averaging this over the forest. DStump [11] simplifies this measure further by only considering the splits made on the root nodes of the trees. We use DStump for global feature selection as part of MAPLE, though MAPLE can be extended to work with other variants.

One of the reasons that local methods are not commonly applied on large-scale problems is that, although their learning rates are minimax optimal, this rate is conservative when not all of the features are involved in the response, as demonstrated empirically in [3]. As a result, we are interested in using a supervised local method. Random forests have two main interpretations towards this: first, as an adaptive method for finding potential nearest neighbors [14] and, second, as a kernel method [19]. Recently, [3] introduced SILO which explicitly uses random forests to define the instance weights for local linear modeling. Empirically, they found that this decreased the bias of the random forest and increased its variance, which is potentially problematic in high dimensional settings.

3 MAPLE

Our proposed method, MAPLE (**M**odel **A**gnostic Su**P**ervised **L**ocal **E**xplanations), combines the idea of using random forests as a method for supervised neighborhood selection for local linear modeling, introduced in [3] as SILO, with the feature selection method proposed in [11] as DStump. For a given point, SILO defines a local neighborhood by assigning a weight to each training point based on how frequently that training point appears in the same leaf node as the given point across the trees in the random forest. DStump defines the importance of a feature based on how much it reduces the impurity of the label when it is split on at the root of the trees in the random forest. These methods are explained in more detail shortly.

Under certain regularity conditions, SILO has been shown to be consistent in that its estimator converges in probability to the true function [3]. Further, DSTump has been shown to identify the active features in a high dimensional setting under the assumption of a general additive model [11]. As a result, the combination of these methods is likely an effective model in a variety of problem settings because it should inherit both of these properties. However, rather than focus on the statistical properties and empirical accuracy of this combination, we focus instead on how the resulting algorithm can be applied to problems where interpretability is a concern.

Before formally defining these procedures and our method, we introduce some notation. Let $x \in \mathbb{R}^{p+1}$ be a feature vector; we assume $[x]_0 = 1$ is a constant term. The index $j \in \{0, \ldots, p\}$ will refer to a specific feature in this vector. Next, let $\{x_i\}_{i=1}^n$ be the training set. The index i will refer to a specific feature vector (training point) in the training set. Then $X \in \mathbb{R}^{n \times (p+1)}$ will denote the matrix representation of the training set (i.e., $[X]_{i,j} = [x_i]_j$). Finally, let $\{T_k\}_{k=1}^K$ be the trees in the random forest; the index k will always refer to a specific tree.

We begin by defining the process by which SILO computes the weights of the training points (i.e., the *local training distribution*) and makes predictions. Let $leaf_k(x)$ be the index of the leaf node of

T_k that contains x. We then define the connection function of the kth tree as

$$c_k(x, x') = \mathbb{1}\{leaf_k(x) = leaf_k(x')\}$$

so the number of training points in the same leaf node as x is

$$num_k(x) = \sum_{i=1}^{n} c_k(x_i, x)$$

. Finally, the weight function of the random forest for the ith training point at the point x is

$$w(x_i, x) = \frac{1}{K} \sum_{k=1}^{K} \frac{c_k(x_i, x)}{num_k(x)}. \tag{2}$$

For a random forest, the model prediction can be written as

$$\hat{f}_{RF}(x) = \frac{1}{K} \sum_{k=1}^{K} \frac{\sum_{i=1}^{n} c_k(x_i, x) y_i}{num_k(x)}. \tag{3}$$

For SILO, the prediction is given by evaluating the solution to the weighted linear regression problem defined by $\{x_i, w(x_i, x), y_i\}$ at x. Let $W_x \in \mathbb{R}^{n \times n}$ be the diagonal weight matrix where $[W_x]_{i,i} = w(x_i, x)$. Then SILO's prediction is

$$\hat{f}_{SILO}(x) = \hat{\beta}_x^T x \text{ where } \hat{\beta}_x = (X^T W_x X)^{-1} X^T W_x y. \tag{4}$$

Next, we define the process DStump uses to select features. Let $split_k \in \{1, \ldots, p\}$ be the index of the feature that the root node of the kth tree split on and suppose that that split reduces the impurity of the label by r_k. Then DStump assigns feature j the score

$$s_j = \sum_{k=1}^{K} \mathbb{1}\{split_k = j\} r_k$$

, and chooses the subset, $A_d \subset \{1, \ldots, p\}$, of the d highest scored features.

MAPLE combines these procedures by using SILO's local training distribution and the best d features from DStump (along with a constant term corresponding to the bias in the local linear model) to solve the weighted linear regression problem $\{[x_i]_{A_d}, w(x_i, x), y_i\}$. Formally, let $Z_d = [X]_{:, A_d} \in \mathbb{R}^{n \times (d+1)}$ and $z_d = [x]_{A_d} \in \mathbb{R}^{d+1}$. Then MAPLE makes the prediction

$$\hat{f}_{MAPLE}(x) = \hat{\beta}_{x,d}^T z_d \text{ where } \hat{\beta}_{x,d} = (Z_d^T W_x Z_d)^{-1} Z_d^T W_x y. \tag{5}$$

Choosing d: We pick the number of features to use in our local linear model via a greedy forward selection procedure that relies on the fact that features can be sorted by their s_j scores. Specifically, we evaluate the predictive accuracy of MAPLE on a held out validation set for $d = 1, \ldots, p$ and choose the value of d that gives the best validation accuracy. It would also be possible to choose d based on the causal local metric.

Extension to Gradient Boosted Regression Trees (GBRT): GBRTs [9], an alternative tree ensemble approach, can also naturally be integrated with MAPLE. Indeed, GBRT tree ensembles can be used to generate local training distributions and feature scores, which can then be fed into MAPLE.

4 MAPLE as an Explanation System

In this section we describe how to use MAPLE to generate explanations ([17] gives details on practical applications of these explanations). The general process of using MAPLE to explain a prediction is essentially the same whether we use MAPLE as a predictive model or only as a black-box explainer; the only difference is that in the first case we fit MAPLE directly on the response variable, while in the second case we fit it on the predictive model's predicted response. MAPLE's local training distribution is vital to these explanations, and is what enables it to address two core weaknesses of local explanations related to 1) diagnosing their limitations in the presence of global patterns, and 2) selecting an appropriate explanation for a new test point when restricted to an existing set of exemplar explanations. We discuss both of these topics in the remainder of this section.
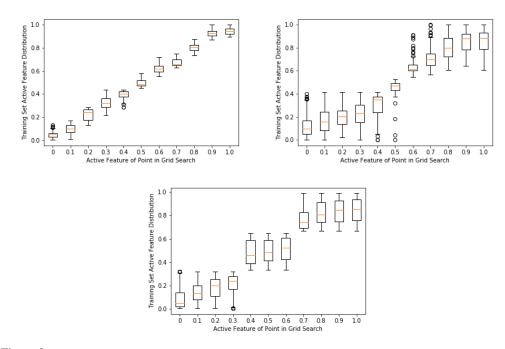
Figure 2: Distribution of the active feature of the most influential points in the training set across a grid search over the range of feature values. (a) **Linear Dataset**: The distributions are roughly uniform in width, typically centered, and change smoothly. (b) **SIL Dataset**: The distributions are wider in the flatter regions of the function and narrower in the transition. They also follow two disjoint ranges with one intermediate range on the transition between the two flatter regions. (c) **Step Dataset**: The distributions are wide, essentially disjoint across the discontinuities of the step function, and not centered.

4.1 Generating Explanations and Detecting Global Patterns

When MAPLE makes a prediction/local explanation, it uses a local linear model, where the coefficients determine the estimated local effect of each feature. If a feature coefficient is non-zero, then we can interpret the impact of the feature according to the sign and magnitude of the coefficient. However, a zero coefficient has two possible interpretations: 1) the feature does not contain global patterns and so it is, indeed, locally inactive; or 2) the feature does contain global patterns and this feature is in fact significant (though not necessarily locally significant). Consequently, our main goal is *diagnosing* the efficacy of a local explanation to determine whether a feature with a zero coefficient contains global patterns. We summarize how we can leverage the local training distribution to do this (see additional discussion in Sec. 5.4).

In particular, we propose two diagnostics for each feature. The first (and simpler) diagnostic involves using the local training distribution for the given test point to create a boxplot to visualize the distribution of each feature. If the boxplot is substantially skewed (i.e., not centered around the test point), then that feature likely contains a global pattern and the test point is nearby it. If the boxplots are not skewed, then the second diagnostic involves performing a grid search across the range of the feature in question. For each value on the grid, we can sample the remaining features in some reasonable way (e.g., by finding several training points with a similar feature value, assuming feature independence and sampling from the empirical distribution, or via MCMC), and create a boxplot for the local training distribution across this grid (see Fig. 2 for examples of this type of plot, the experimental setup is described in Sec. 5.4). If the local training distributions appear to share similar boundaries that change abruptly during the grid search, as seen in Fig. 2c and somewhat in Fig. 2b, then there is likely a global pattern present in that feature. Conversely, if the local training distributions are roughly centered around the test point during the grid search and change smoothly during it, as seen in Fig. 2a, then the effect of that feature likely does not have significant global patterns.

4.2 Picking an Exemplar Explanation

As noted in [17], there are settings where we want to compile a small set of (presumably diverse) representative exemplar explanations, and use these exemplar explanations to explain new test points. Selecting the appropriate exemplar for a new test point is a challenge for existing local explanation systems [18]. MAPLE provides an elegant solution to this problem by using the local training distributions. Specifically, we can determine if we can apply a particular exemplar explanation to a proposed test point by evaluating how likely the proposed point is under the exemplar explanation's local training distribution.

Of course, there is no guarantee that, collectively, these distributions span the entire input space. If asked to explain a test point from an uncovered part of the input space, then the proposed point will have low probability under all of the exemplar explanation distributions, and, having noticed that, we can determine that no exemplar explanation should be applied. Similarly, we can detect if multiple exemplar explanations may be equally applicable.

5   Experimental Results

We present experiments demonstrating that: 1) MAPLE is generally at least as accurate as random forests, GBRT, and SILO 2) MAPLE provides faithful self-explanations, i.e., its local linear model at x is a good local explanation of the prediction at x 3) MAPLE is more accurate in predicting a black-box predictive model's response than a comparable and popular explanation system, LIME [17] 4) The local training distribution can be used to detect the presence of global patterns in the predictive model.

5.1   Accuracy on UCI datasets

We run our experiments on several of the UCI datasets [7]. Each dataset was divided into a 50/25/25 training, validation, and testing split for each of the trials. All variables, including the response, were standardized to have mean zero and variance one.

We compare MAPLE and SILO with tree ensembles constructed using standard random forests (RF) and gradient boosted regression trees (GBRT). The ensemble choice for the baseline impacts the structure of the trees, which alters the weights as well as the global features selected by DStump. We include the performance of a linear model (LM) as well. Root mean squared errors (RMSE) are reported in Table 1 and the number of selected features is in Table 4. Overall, MAPLE does at least as well as the tree ensembles and SILO, and often does better (the Music dataset being the sole exception).

5.2   Faithful Self-Explanations

We next demonstrate that the local linear model that MAPLE uses to make its prediction doubles as an effective local explanation. The general data processing is the same as in the previous section, and we restrict our results to the random forest based version of MAPLE. We use our proposed causal metric defined in (1) as our evaluation metric, defining p_x as $\mathcal{N}(x, \sigma I)$, using the squared l_2 loss, and approximating the expectation by taking x from the testing set and drawing five x' per testing point.

We chose $\sigma = 0.1$ as a reasonable choice for the neighborhood scale because the data was normalized to have variance one. The results, which show the RMSE of the causal metric in Table 2, demonstrate that the local linear models produced are good local explanations for the model as a whole when compared to using LIME to explain MAPLE.

5.3   MAPLE as a Black-box Explainer

The overall setup for these experiments is the same as in the previous section, except that we are evaluating explanation systems in the black-box setting, where they are fit against the predictive model's predicted response. We use a Support Vector Regression (SVR) model (implementation and standard parameters from scikit-learn) as a black-box predictive model. We present a comparison of MAPLE to LIME in Table 3. Our results show the RMSE of the causal metric. MAPLE produces

Dataset	LM	RF	SILO + RF	MAPLE + RF	GBRT	SILO + GBRT	MAPLE + GBRT
Autompgs	0.446	0.4164	0.3784	0.381	0.392	0.3745	0.377
Communities	0.781	0.745	0.724	**0.688**	0.709	0.751	**0.712**
Crimes	0.327	1.012	0.531	**0.331**	0.968	0.493	**0.295**
Day	0	0.204	1.7e-05	**6e-06**	0.104	1.3e-05	**4e-06**
Happiness	0.001	0.644	0.001	0.001	0.344	0.001	0.001
Housing	0.56	0.486	0.409	0.419	0.395	0.396	0.404
Music	0.935	0.742	0.881	**0.764**	0.658	0.901	**0.849**
Winequality-red	0.814	0.78	0.779	0.778	0.783	0.786	0.779

Table 1:   Average RMSE across 50 trials; underlined results indicate that MAPLE differed significantly from the baseline method (RF or GBRT) and bold results indicate that MAPLE differed significantly from SILO built on the same baseline. With the exception of the Music dataset, MAPLE is at least as good as the baseline.

Dataset	LIME	MAPLE
Autompgs	0.178	**0.042**
Communities	0.409	**0.130**
Crimes	0.276	**0.047**
Day	0.034	**0**
Happiness	0.05	**3e-05**
Housing	0.238	**0.07**
Music	0.189	0.181
Winequality-red	0.149	**0.06**

Table 2:   A comparison of the causal metric of MAPLE vs LIME for $\sigma = 0.1$ when being used to explain the predictions of MAPLE. Values shown are RMSE averaged over 25 trials. Bold entries denote a significant difference.

Dataset	SVR	LIME	MAPLE
Autompgs	0.39	0.282	**0.15**
Communities	0.761	0.338	**0.323**
Crimes	0.895	0.183	0.232
Day	0.2	0.242	**0.17**
Happiness	0.267	0.28	**0.187**
Housing	0.459	0.366	**0.206**
Music	0.816	0.326	**0.304**
Winequality-red	0.807	0.295	**0.204**

Table 3:   A comparison of the causal metric of MAPLE vs LIME for $\sigma = 0.1$ when being used in the black-box setting to explain the predictions of a SVR model. Values shown are RMSE averaged over 25 trials. Bold entries denote a significant difference.

Dataset	n	p	d - RF	d - GBRT
Autompgs	392	8	6.44	5.94
Communities	1993	103	54.14	50.12
Crimes	2214	103	20.34	21.62
Day	731	15	2.46	3.02
Happiness	578	8	7.74	7.46
Housing	506	12	9.98	10.06
Music	1059	70	5.56	14.46
Winequality-red	1599	12	7.1	6.88

Table 4:   For each dataset, its size, n, and dimension, p, along with the average number of features used, d, by MAPLE for RF and GBRT respectively.

more accurate local explanations than LIME for all but the Crimes dataset, where the difference was not statistically significant.

What about a Larger σ?: After data normalization, the median range of our features (across all datasets) is roughly six. For a single dimension, the width of the 95% confidence interval for the sample used in the causal metric is approximately 4σ. In addition to the reported results with $\sigma = 0.1$, we ran experiments with $\sigma = 0.25$. For this larger value, the expected range of the neighborhood is approximately one sixth of the overall feature range. Further, the probability that one dimension falls outside of this interval increases exponentially with the number of dimensions. Consequently, the neighborhoods are even larger for high dimensional problems. Thus $\sigma = 0.25$ appears to be unreasonably large for a local explanation in high dimensions, especially considering that standard nearest neighbor methods frequently rely on a small constant number of neighbors. Nonetheless, we note that for $\sigma = 0.25$, we found that MAPLE significantly outperformed LIME on the Autompgs, Happiness, Housing, and Winequality-red datasets. In contrast, LIME significantly outperformed MAPLE on the three datasets with largest dimension p (see Table 4): Communities, Crimes, and Music.

5.4   Using Influential Training Points

To demonstrate how the local training distribution can be used to make inferences about the global patterns of the model, we work with the datasets introduced in Fig. 1 of Sec. 1. Each dataset consists

of $n = 200$ draws from $[0, 1]^5$ taken uniformly at random. These samples are passed through either a linear, shifted inverse logistic (SIL), or step function that acts only on the first dimension of the feature vector (the remaining four dimensions are noisy features), and then a normally distributed noise is added with $\sigma = 0.1$. We refer to the first feature as the 'active' feature. We then fit a random forest to the data and fit MAPLE to the random forest's predicted response. Next, we do a grid search across the active feature (e.g., $[x]_0 = 0, 0.1, \ldots, 1.0$) and sample the remaining features uniformly at random from $[0, 1]$ (simulating sampling the remaining features from the data distribution). For a given sampled point x', we use MAPLE's local training distribution to identify the 20 most influential training points and plot a boxplot of the distribution of the active feature for these influential points. To smooth the results, we repeat this procedure 10 times for each point in the grid search. The plots of these distributions are in Fig. 2.

Interpreting these Distributions: When a random forest fits a continuous function, each tree splits the input space into finer and finer partitions and, if the slope of the function does not change too rapidly, the distribution of where the partitions are split are relatively uncorrelated between the trees. As a result, the influential training points for a prediction at x are roughly centered around x and tend to change smoothly as x changes. This is demonstrated in Fig. 2a with the exception of points near the boundary of the distribution.

Further, the steeper the function is at x, the finer the partitions become around x and, consequently, the distribution of the influential points becomes more concentrated. This can be seen by contrasting Fig. 2a and Fig. 2b; in Fig. 2a, the distributions have roughly equal variance while, in Fig. 2b, the distributions in the flat areas of the function have large variance while the distributions in the steep transition of the function are much narrower. Although not shown, one extreme of this is a feature that is inactive; in all of our experiments, the inactive feature distributions of the most influential training points look like the original data distribution.

In the extreme case, when fitting a discontinuous function, the trees are likely to all split at or near the discontinuity. As a result, influential training points for x may not be centered around x when x is near the discontinuity and they will change abruptly as x moves past the discontinuity. This is demonstrated in Fig. 2c where we clearly observe the transitions of the step function, as well as the fact that the intervals are not centered around x. Further, in Fig. 2b we can see the fact that there are two flat areas of the function with a steep and short transition between by noticing that there are two main ranges for the influential points with one intermediate range at $x_0 = 0.5$.

6   Conclusion and Future Work

We have shown that MAPLE is effective both as a predictive model and an explanation system. Additionally, we have demonstrated how to use its local training distribution to address two key weaknesses of local explanations: 1) Detecting and modeling global patterns, and 2) Determining whether an exemplar explanation can be applied to a new test point. Some interesting avenues of future work include: 1) Exploiting the fact that MAPLE is a locally linear model to tap into the wide range of approaches that use influence functions to improve model accuracy or identify interesting data points via measures such as leverage and Cook's distance [5]; 2) Exploring the use of local feature selection approaches with MAPLE, e.g., by considering impurity reductions along the paths through all trees for a given test point; and 3) Exploring methods other than tree ensembles for defining the similarity weights from Eq. 2.

Acknowledgements

References

[1] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.

[2] Jacob Bien and Robert Tibshirani. Prototype selection for interpretable classification. *The Annals of Applied Statistics*, pages 2403–2424, 2011.

[3] Adam Bloniarz, Ameet Talwalkar, Bin Yu, and Christopher Wu. Supervised neighborhoods for distributed nonparametric regression. In *Artificial Intelligence and Statistics*, pages 1450–1459, 2016.

[4] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[5] Samprit Chatterjee and Ali S Hadi. Influential observations, high leverage points, and outliers in linear regression. *Statistical Science*, pages 379–393, 1986.

[6] R Dennis Cook and Sanford Weisberg. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics*, 22(4):495–508, 1980.

[7] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017.

[8] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. 2017.

[9] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[10] Shachar Kaufman, Saharon Rosset, Claudia Perlich, and Ori Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):15, 2012.

[11] Jalil Kazemitabar, Arash Amini, Adam Bloniarz, and Ameet S Talwalkar. Variable importance using decision trees. In *Advances in Neural Information Processing Systems*, pages 425–434, 2017.

[12] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730*, 2017.

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

[14] Yi Lin and Yongho Jeon. Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101(474):578–590, 2006.

[15] Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[16] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4768–4777, 2017.

[17] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. AAAI, 2018.

[19] Erwan Scornet. Random forests and kernel methods. *IEEE Transactions on Information Theory*, 62(3):1485–1500, 2016.