Explaining the Explainer: A First Theoretical Analysis of LIME

Damien Garreau, Ulrike von Luxburg

Explaining the Explainer: A First Theoretical Analysis of LIME

Damien Garreau[1,3]
damien.garreau@unice.fr

Ulrike von Luxburg[1,2]
ulrike.luxburg@uni-tuebingen.de

[1]Max Planck Institute for Intelligent Systems, Germany
[2]University of Tübingen, Germany
[3]Université Côte d'Azur, Inria, CNRS, LJAD, France

Abstract

Machine learning is used more and more often for sensitive applications, sometimes replacing humans in critical decision-making processes. As such, interpretability of these algorithms is a pressing need. One popular algorithm to provide interpretability is LIME (Local Interpretable Model-Agnostic Explanation). In this paper, we provide the first theoretical analysis of LIME. We derive closed-form expressions for the coefficients of the interpretable model when the function to explain is linear. The good news is that these coefficients are proportional to the gradient of the function to explain: LIME indeed discovers meaningful features. However, our analysis also reveals that poor choices of parameters can lead LIME to miss important features.

1 Introduction

1.1 Interpretability

The recent advance of machine learning methods is partly due to the widespread use of very complicated models, for instance deep neural networks. As an example, the Inception Network (Szegedy et al., 2015) depends on approximately 23 million parameters. While these models achieve and sometimes surpass human-level performance on certain tasks (image classification being one of the most famous), they are often perceived as *black boxes*, with little understanding of how they make individual predictions.

This lack of understanding is a problem for several reasons. First, it can be a source of catastrophic errors when these models are deployed *in the wild*. For instance, for any safety system recognizing cars in images, we want to be absolutely certain that the algorithm is using features related to cars, and not exploiting some artifacts of the images. Second, this opacity prevents these models from being *socially accepted*. It is important to get a basic understanding of the decision making process to accept it.

Model-agnostic explanation techniques aim to solve this interpretability problem by providing qualitative or quantitative help to understand how black-box algorithms make decisions. Since the global complexity of the black-box models is hard to understand, they often rely on a *local* point of view, and produce an interpretation for a specific instance. In this article, we focus on such an explanation technique: **Local Interpretable Model-Agnostic Explanations** (LIME, Ribeiro et al. (2016)).

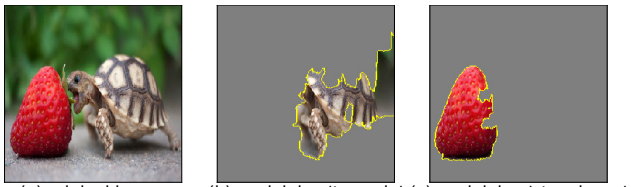

Figure 1: LIME explanation for object identification in images. We used Inception (Szegedy et al., 2015) as a black-box model. Terrapin, a sort of turtle, is the top label predicted for the image in panel (a). Panel (b) shows the results of LIME, explaining how this prediction was made. The highlighted parts of the image are the superpixels with the top 5 coefficients in the surrogate linear model. We ran the same experiment for the 'strawberry' label in panel (c).

1.2 Contributions

Our main goal in this paper is to provide theoretical guarantees for LIME. On the way, we shed light on

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

some interesting behaviors of the algorithm in a simple setting. Our analysis is based on the Euclidean version of LIME, called "tabular LIME." Our main results are the following:

(i). When the model to explain is linear, we **compute in closed-form** the average coefficients of the surrogate linear model obtained by `TabularLIME`.
(ii). In particular, these coefficients are **proportional to the partial derivatives of the black-box model** at the instance to explain. This implies that `TabularLIME` indeed highlights important features.
(iii). On the negative side, using the closed-form expressions we show that **it is possible to make some important features disappear** in the interpretation, just by changing a parameter of the method.
(iv). We also compute the local error of the surrogate model, and show that it is **bounded** away from 0 in general.

We explain how `TabularLIME` works in more details in Section 2. In Section 3, we state our main results. They are discussed in Section 4, and we provide an outline of the proof of our main result in Section 5. We conclude in Section 6.

2 LIME: Outline and notation

2.1 Intuition

From now on, we will consider a particular model encoded as a function $f : \mathbb{R}^d \to \mathbb{R}$ and a particular instance $\xi \in \mathbb{R}^d$ to explain. We make no assumptions on this function, *e.g.*, how it might have been learned. We simply consider f as a black-box model giving us predictions for all points of the input space. Our goal will be to explain the decision $f(\xi)$ that this model makes for one particular instance ξ.

As soon as f is too complicated, it is hopeless to try and fit an interpretable model globally, since the interpretable model will be too simple to capture all the complexity of f. Thus a reasonable course of action is to consider a *local* point of view, and to explain f in the neighborhood of some fixed instance ξ. This is the main idea behind LIME: To explain a decision for some fixed input ξ, sample other examples around ξ, use these samples to build a simple interpretable model in the neighborhood of ξ, and use this surrogate model to explain the decision for ξ.

One additional idea that makes a huge difference with other existing methods is to use *discretized* features of smaller dimension d' to build the local model. These new categorical features are easier to interpret, since they are categorical. In the case of images, they are built by using a split of the image ξ into superpixels (Ren and Malik, 2003). See Figure 1 for an example of LIME output in the case of image classification. In this situation, the surrogate model highlights the superpixels of the image that are the most "active" in predicting a given label.

Whereas LIME is most famous for its results on images, it is easier to understand how it operates and to analyze theoretically on **tabular data**. In the case of tabular data, LIME works essentially in the same way, with a main difference: tabular LIME requires a train set, and each feature is discretized according to the empirical quantiles of this training set.

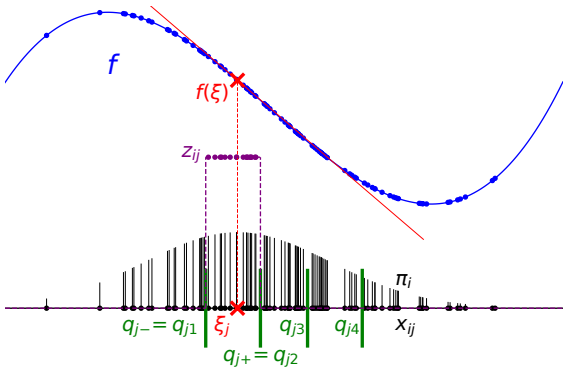

Figure 2: General setting of `TabularLIME` along coordinate j. Given a specific datapoint ξ (in red), we want to build a local model for f (in blue), given new samples $x_1, \dots, x_n$ (in black). Discretizing with respect to the quantiles of the distribution (in green), these new samples are transformed into categorical features z_i (in purple). In the construction of the surrogate model, they are weighted with respect to their proximity with ξ (here exponential weights given by Eq. (2.1), in black). In red, we plotted the tangent line, the best linear approximation one could hope for.

We now describe the general operation of LIME on Euclidean data, which we call `TabularLIME`. We provide synthetic description of `TabularLIME` in Algorithm 1, and we refer to Figure 2 for a depiction of our setting along a given coordinate. Suppose that we want to explain the prediction of the model f at the instance ξ. `TabularLIME` has an intricate way to sample points in a local neighborhood of ξ. First, `TabularLIME` constructs empirical quantiles of the train set on each dimension, for a given number p of bins. These quantile boxes are then used to construct a discrete representation of the data: if ξ_j falls between $\hat{q}_k$ and $\hat{q}_{k+1}$, it receives the value k. We now have a discrete version of ξ, say $(2, 3, \dots)^\top$. The next step is to sample discrete examples in $\{1, \dots, p\}^d$ uniformly at random: for instance, $(1, 3, \dots)^\top$ means that `TabularLIME` sampled an encoding such that the first coordinate falls into

the first quantile box, the second coordinate into the third, etc. `TabularLIME` subsequently un-discretizes these encodings by sampling from a normal distribution truncated to the corresponding quantile boxes, obtaining *new examples* $x_1, \dots, x_n$. For example, for sample $(1, 3, \dots)^\top$ we now sample the first coordinate from a normal distribution restricted to quantile box #1, the second coordinate from quantile box #3, etc. This sampling procedure ensures that we have samples in each part of the space. The next step is to convert these sampled points to binary features, indicating for each coordinate if the new example falls into the same quantile box as ξ. Here, z_i would be $(1, 0, \dots)^\top$. Finally, an interpretable model (say linear) is learned using these binary features.

Algorithm 1 `TabularLIME` for regression

Require: Model f, # of new samples n, instance ξ, bandwidth ν, # of bins p, mean μ, variance σ^2

$q \leftarrow$ `GetQuantiles`(p,μ,σ)
$t \leftarrow$ `Discretize`(ξ,q)
for $i = 1$ to n **do**
  for $j = 1$ to d **do**
    $y_{i,j} \leftarrow$ `SampleUniform`$(\{1, \dots, p\})$
    $(q_\ell, q_u) \leftarrow (q_{j,y_{ij}}, q_{j,y_{ij}+1})$
    $x_{i,j} \leftarrow$ `SampleTruncGaussian`$(q_\ell, q_u, \mu, \sigma)$
    $z_{i,j} \leftarrow \mathbf{1}_{t_j = y_{i,j}}$
  end for
  $\pi_i \leftarrow \exp\left(\frac{-\|x_i - \xi\|^2}{2\nu^2}\right)$
end for
$\widehat{\beta} \leftarrow$ `WeightedLeastSquares`$(z, f(x), \pi)$
return $\widehat{\beta}$

2.2 Implementation choices and notation

LIME is a quite general framework and leaves some freedom to the user regarding each brick of the algorithm. We now discuss each step of `TabularLIME` in more detail, presenting our implementation choices and introducing our notation on the way.

Discretization. As said previously, the first step of `TabularLIME` is to create a partition of the input space using a train set. Intuitively, `TabularLIME` produces *interpretable features* by discretizing each dimension. Formally, given a fixed number of bins p, for each feature j, the empirical quantiles $\hat{q}_{j,0}, \dots, \hat{q}_{j,p}$ are computed. Thus, along each dimension, there is a mapping $\hat{\phi}_j : \mathbb{R} \to \{1, \dots, p\}$ associating each real number to the index of the quantile box it belongs to. For any point $x \in \mathbb{R}^d$, the interpretable features are then defined as a $0-1$ vector corresponding to the discretization of x being the same as the discretization of ξ. Namely, $z_j = \mathbf{1}_{\hat{\phi}_j(x) = \hat{\phi}_j(\xi)}$ for all $1 \leqslant j \leqslant d$. Intuitively, these categorical features correspond to the *absence* or *presence* of interpretable components. The discretization process makes a huge difference with respect to other methods: we lose the obvious link with the gradient of the function, and it is much more complicated to see how the local properties of f influence the result of the LIME algorithm, even in a simple setting. In all our experiments, we took $p = 4$ (quartile discretization, the default setting).

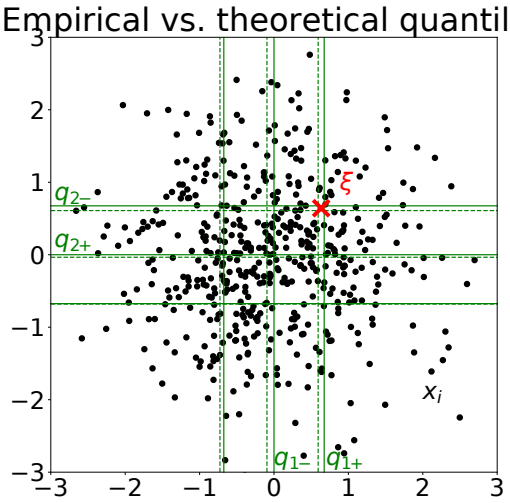

Figure 3: A visualization of the train set in dimension $d = 2$ with $\mu = (0, 0)^\top$, and $\sigma^2 = 1$. The empirical quantiles (dashed green lines) are already very close to the theoretical quantiles (green lines) for $n_{\text{train}} = 500$. The main difference in the procedure appears if ξ (red cross) is chosen at the edge of a quantile box, changing the way all the new samples are encoded. But for a train set containing enough observations and a generic ξ, there is virtually no difference between using the theoretical quantiles and the empirical quantiles.

Sampling strategy. Along with $\hat{\phi}$, `TabularLIME` creates an un-discretization procedure $\hat{\psi} : \{1, \dots, p\} \to \mathbb{R}$. Simply put, given a coordinate j and a bin index k, $\hat{\psi}_j(k)$ samples a truncated Gaussian on the corresponding bin, with parameters computed from the training set. The `TabularLIME` sampling strategy for a new example amounts to (i) sample $y_i \in \{1, \dots, p\}^d$ a random variable such that the y_{ij} are independent samples of the discrete uniform distribution on $\{1, \dots, p\}$, and (ii) apply the un-discretization step, that is, return $\hat{\psi}(y)$. We will denote by $x_1, \dots, x_n \in \mathbb{R}^d$ these new examples, and $z_1, \dots, z_n \in \{0, 1\}^d$ their discretized counterparts. Note that it is possible to take other bin boxes than those given by the empirical quantiles, the y_{ij}s are then sampled according to the frequency observed in the dataset. The sampling step of `TabularLIME` helps to explore the values of the function in the neighborhood of the instance to explain. Thus it is not so important to sample according to the distribution of the data, and a Gaussian sampling that mimics it is enough.

Assuming that we know the distribution of the train data, it is possible to use the theoretical quantiles instead of the empirical ones. For a large number of examples, they are arbitrary close (see, for instance, Lemma 21.2 in Van der Vaart (2000)). See Figure 3 for an illustration. It is this approach that we will take from now on: we denote the discretization step by ϕ and denote the quantiles by q_{jk} for $1 \leqslant j \leqslant d$ and $0 \leqslant k \leqslant p$ to mark this slight difference. Also note that, for every $1 \leqslant j \leqslant d$, we set $q_{j\pm}$ the quantiles bounding ξ_j, that is, $q_{j-} \leqslant \xi_j < q_{j+}$ (see Figure 2).

Train set. `TabularLIME` requires a train set, which is left free to the user. In spirit, one should sample according to the distribution of the train set used to fit the model f. Nevertheless, this train set is rarely available, and from now on, we choose to consider draws from a $\mathcal{N}\left(\mu, \sigma^2 \mathrm{I}_d\right)$. The parameters of this Gaussian can be estimated from the training data that was used for f if available. Thus, in our setting, along each dimension j, the $(q_{jk})_{0\leqslant k\leqslant p}$ are the (rescaled) quantiles of the normal distribution. In particular, they are identical for all features. A fundamental consequence is that sampling the new examples x_is first and then discretizing **has the same distribution** as sampling first the bin indices y_is and then un-discretizing.

Weights. We choose to give each example weight

$$\pi_i := \exp\left(\frac{-\left\|x_i - \xi\right\|^2}{2\nu^2}\right), \tag{2.1}$$

where $\|\cdot\|$ is the Euclidean norm on $\mathbb{R}^d$ and $\nu > 0$ is a bandwidth parameter. It should be clear that ν is a hard parameter to tune:

- if ν is very large, then **all the examples receive positive weights:** we are trying to build a simple model that captures the complexity of f at a global scale. This cannot work if f is too complicated.
- if ν is too small, then **only examples in the immediate neighborhood of** ξ receive positive weights. Given the discretization step, this amounts to choosing $z_i = (1, \ldots, 1)^\top$ for all i. Thus the linear model built on top would just be a constant fit, missing all the relevant information.

Note that other distances than the Euclidean distance can be used, for instance the cosine distance for text data. The default implementation of LIME uses $\|z_i - t\|$ instead of $\|x_i - \xi\|$, with bandwidth set to $0.75d$. We choose to use the true Euclidean distance between ξ and the new examples as it can be seen as a smoothed version of the distance to z_i and has the same behavior.

Interpretable model. The final step in `TabularLIME` is to build a local interpretable model. Given a class of simple, interpretable models G, `TabularLIME` selects the best of these models by solving

$$\operatorname*{arg\,min}_{g\in G}\left\{L_n(f, g, \pi_\xi) + \Omega\left(g\right)\right\}, \tag{2.2}$$

where L_n is a local loss function evaluated on the new examples $x_1, \ldots, x_n$, and $\Omega : \mathbb{R}^d \to \mathbb{R}$ is a regularizer function. For instance, a natural choice for the local loss function is the weighted squared loss

$$L_n(f, g, \pi) := \frac{1}{n}\sum_{i=1}^{n} \pi_i \left(f(x_i) - g(z_i)\right)^2. \tag{2.3}$$

We saw in Section 1.1 different possibilities for G. In this paper, we will focus exclusively on the linear models, in our opinion the easiest models to interpret. Namely, we set $g(z_i) = \beta^\top z_i + \beta_0$, with $\beta \in \mathbb{R}^d$ and $\beta_0 \in \mathbb{R}$. To get rid of the intercept β_0, we now use the standard approach to introduce a phantom coordinate 0, and $z, \beta \in \mathbb{R}^{d+1}$ with $z_0 = 1$ and $\beta_0 = \beta_0$. We also stack the z_is together to obtain $Z \in \{0, 1\}^{n\times(d+1)}$.

The regularization term $\Omega(g)$ is added to insure further interpretability of the model by reducing the number of non-zero coefficients in the linear model given by `TabularLIME`. Typically, one uses L^2 regularization (ridge regression is the default setting of LIME) or L^1 regularization (the Lasso). To simplify the analysis, we will set $\Omega = 0$ in the following. We believe that many of the results of Section 3 stay true in a regularized setting, especially the switch-off phenomenon that we are going to describe below: coefficients are even more likely to be set to zero when $\Omega \neq 0$.

In other words, in our case `TabularLIME` performs *weighted linear regression* on the interpretable features z_is, and outputs a vector $\widehat{\beta} \in \mathbb{R}^{d+1}$ such that

$$\widehat{\beta} \in \operatorname*{arg\,min}_{\beta\in\mathbb{R}^{d+1}}\left\{\frac{1}{n}\sum_{i=1}^{n} \pi_i (y_i - \beta^\top z_i)^2\right\}. \tag{2.4}$$

Note that $\widehat{\beta}$ is a random quantity, with randomness coming from the sampling of the new examples $x_1, \ldots, x_n$. It is clear that from a theoretical point of view, a big hurdle for the theoretical analysis is the discretization process (going from the x_is to the z_is).

Regression vs. classification. To conclude, let us note that `TabularLIME` can be used both for regression and classification. Here we focus on the *regression* mode: the outputs of the model are real numbers, and not discrete elements. In some sense, this is a more general setting than the classification case, since the classification mode operates as `TabularLIME` for

regression, but with f chosen as the function that gives the likelihood of belonging to a certain class according to the model.

2.3 Related work

Let us mention a few other model-agnostic methods that share some characteristics with LIME. We refer to Guidotti et al. (2019) for a thorough review.

Shapley values. Following Shapley (1953) the idea is to estimate for each subset of features S the expected prediction difference $\Delta(S)$ when the value of these features are *fixed* to those of the example to explain. The contribution of the jth feature is then set to an average of the contribution of j over all possible coalitions (subgroups of features not containing j). They are used in some recent interpretability work, see Lundberg and Lee (2017) for instance. It is extremely costly to compute, and does not provide much information as soon as the number of features is high. Shapley values share with LIME the idea of quantifying how much a feature contributes to the prediction for a given example.

Gradient methods. Also related to LIME, *gradient-based* methods as in Baehrens et al. (2010) provide local explanations without knowledge of the model. Essentially, these methods compute the partial derivatives of f at a given example. For images, this can yield satisfying plots where, for instance, the contours of the object appear: a *saliency map* (Zeiler and Fergus, 2014). Shrikumar et al. (2016, 2017) propose to use the "input × derivative" product, showing advantages over gradient methods. But in any case, the output of these gradient based methods is not so interpretable since the number of features is very high. LIME gets around this problem by using a local dictionary with much smaller dimensionality than the input.

3 Theoretical value of the coefficients of the surrogate model

We are now ready to state our main result. Let us denote by $\widehat{\beta}$ the coefficients of the linear surrogate model obtained by `TabularLIME`. In a nutshell, when the underlying model f is linear, we can derive the average value β of the $\widehat{\beta}$ coefficients. In particular, we will see that the β_js are proportional to the partial derivatives $\partial_j f(\xi)$. The exact form of the proportionality coefficients is given in the formal statement below, it essentially depends on the scaling parameters

$$\tilde{\mu} := \frac{\nu^2\mu + \sigma^2\xi}{\nu^2+\sigma^2} \in \mathbb{R}^d \text{ and } \tilde{\sigma} := \frac{\nu^2\sigma^2}{\nu^2+\sigma^2} > 0\,,$$

and the $q_{j\pm}$s, the quantiles left and right of the ξ_js.

Theorem 3.1 (Coefficients of the surrogate model, theoretical values). *Assume that f is of the form $x \mapsto a^\top x + b$, and set*

$$\beta := \begin{pmatrix} f(\tilde{\mu}) + \sum_{j=1}^d \frac{a_j\theta_j}{1-\alpha_j} \\ \frac{-a_1\theta_1}{\alpha_1(1-\alpha_1)} \\ \vdots \\ \frac{-a_d\theta_d}{\alpha_d(1-\alpha_d)} \end{pmatrix} \in \mathbb{R}^{d+1}\,, \tag{3.1}$$

where, for any $1 \leqslant j \leqslant d$, we defined

$$\alpha_j := \left[\frac{1}{2}\mathrm{erf}\left(\frac{x-\tilde{\mu}_j}{\tilde{\sigma}\sqrt{2}}\right)\right]_{q_{j-}}^{q_{j+}},$$

and

$$\theta_j := \left[\frac{\tilde{\sigma}}{\sqrt{2\pi}}\exp\left(\frac{-(x-\tilde{\mu}_j)^2}{2\tilde{\sigma}^2}\right)\right]_{q_{j-}}^{q_{j+}}.$$

Let $\eta \in (0,1)$. Then, with high probability greater than $1-\eta$, it holds that

$$\left\|\widehat{\beta}-\beta\right\| \lesssim \max(\sigma\left\|\nabla f\right\|, f(\tilde{\mu}) + \tilde{\sigma}\left\|\nabla f\right\|)\sqrt{\frac{\log 1/\eta}{n}}\,.$$

A precise statement with the accurate dependencies in the dimension and the constants hidden in the result can be found in the Appendix (Theorem 10.1). Before discussing the consequences of Theorem 3.1 in the next section, remark that since ξ is encoded by $(1,1,\ldots,1)^\top$, the prediction of the local model at ξ, $\hat{f}(\xi)$, is just the sum of the $\widehat{\beta}_j$s. According to Theorem 3.1, $\hat{f}(\xi)$ will be close to this value, with high probability. Thus we also have a statement about the error made by the surrogate model in ξ.

Corollary 3.1 (Local error of the surrogate model). *Let $\eta \in (0,1)$. Then, under the assumptions of Theorem 3.1, with probability greater than $1-\eta$, it holds that*

$$\left|\hat{f}(\xi) - f(\tilde{\mu}) + \sum_{j=1}^d \frac{a_j\theta_j}{\alpha_j}\right| \leqslant$$
$$\leqslant \max(\sigma\left\|\nabla f\right\|, f(\tilde{\mu}) + \tilde{\sigma}\left\|\nabla f\right\|)\sqrt{\frac{\log 1/\eta}{n}}\,,$$

with hidden constants depending on d and the α_js.

Obviously the goal of `TabularLIME` is not to produce a very accurate model, but to provide interpretability. The error of the local model can be seen as a hint about how reliable the interpretation might be.

4 Consequences of our main results

We now discuss the consequences of Theorem 3.1 and Corollary 3.1.

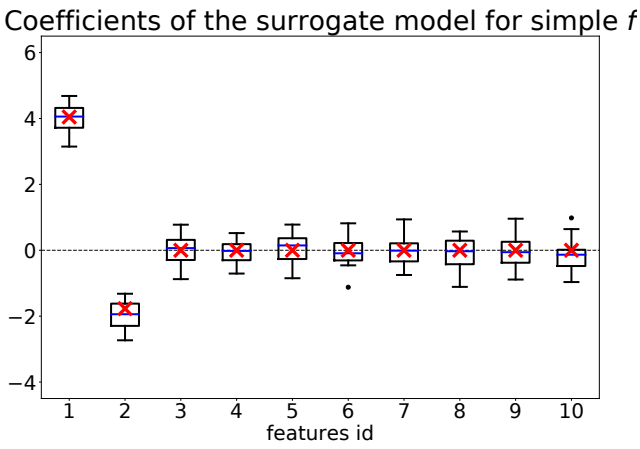

Figure 4: Example where the true underlying black box model only depends on *two* features: $f(x) = 10x_1 - 10x_2$. For each of the 10 features, we plot the values of the $\widehat{\beta}_j$s obtained by `TabularLIME`. The blue line shows the median over all experiments, the red cross the β_j theoretical value according to our theorem. The boxplots contain values between first and third quartiles, the whiskers are 1.5 times the interquartile ranges, and the black dots mark values outside this range. To produce the figure, we made 20 repetitions of the experiment, with $n = 10^4$ examples and $\nu = 1$. We see that `TabularLIME` finds nonzero coefficients exactly for the first two coordinates, up to noise coming from the sampling. This is the result that one would hope to achieve, and also the result predicted by our theory.

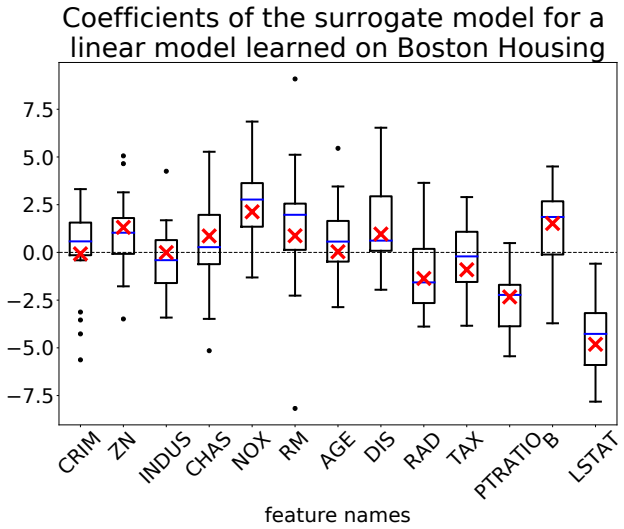

Figure 5: Values of the coefficients obtained by `TabularLIME` on each coordinate in dimension $d = 13$ for a linear model trained on the Boston housing dataset (Harrison Jr. and Rubinfeld, 1978). The β_js are concentrated around the red crosses, which denote the β_js, the theoretical values predicted by Theorem 3.1. To produce the figure, we ran 20 experiments with $n = 10^3$ new samples generated for each run and we set $\nu = 1$.

Dependency in the partial derivatives. A first consequence of Theorem 3.1 is that the coefficients of the linear model given by `TabularLIME` are approximately **proportional to the partial derivatives of** f at ξ, with constant depending on our assumptions. An interesting follow-up is that, if f depends only on a few features, then the partial derivatives in the other coordinates are zero, and the coefficients given by `TabularLIME` for these coordinates will be 0 as well. For instance, if $f(x) = 10x_1 - 10x_2$ as in Figure 4, then $\beta_1 \simeq 11.4$, $\beta_2 \simeq -4.1$, and $\beta_j = 0$ for all $j \geqslant 3$. In a simple setting, we thus showed that `TabularLIME` does not produce interpretations with additional erroneous feature dependencies. Indeed, when the number of samples is high, the coordinates which do not influence the prediction will have a coefficient close to the theoretical value 0 in the surrogate linear model. For a bandwidth not too large, this dependency in the partial derivatives seems to hold to some extent for more general functions. See for instance Figure 6, where we demonstrate this phenomenon for a kernel regressor.

Robustness of the explanations. Theorem 3.1 means that, for large n, `TabularLIME` outputs coefficients that are very close to β with high probability, where β is a vector that can be computed explicitly as per Eq. (3.1). Still without looking too closely at the values of β, this is already interesting and hints that there is some robustness in the interpretations provided by `TabularLIME`: given enough samples, the explanation will not jump from one feature to the other. This is a desirable property for any interpretable method, since the user does not want explanations to change randomly with different runs of the algorithm. We illustrate this phenomenon in Figure 5.

Influence of the bandwidth. Unfortunately, Theorem 3.1 does not provide directly a founded way to pick ν, which would for instance minimize the variance for a given level of noise. The quest for a founded heuristic is still open. However, we gain some interesting insights on the role of ν. Namely, for fixed ξ, μ, and σ, the multiplicative constants $\theta_j/(\alpha_j(1-\alpha_j))$ appearing in Eq. (3.1) depend essentially on ν.

Without looking too much into these constants, one can already see that they regulate the magnitude of the coefficients of the surrogate model in a non-trivial way. For instance, in the experiment depicted in Figure 4, the partial derivative of f along the two first coordinate has the same magnitude, whereas the interpretable coefficient is much larger for the first coordinate than the second. Thus we believe that the value of the coefficients in the obtained linear model should not be taken too much into account.

More disturbing, it is possible to artificially (or by

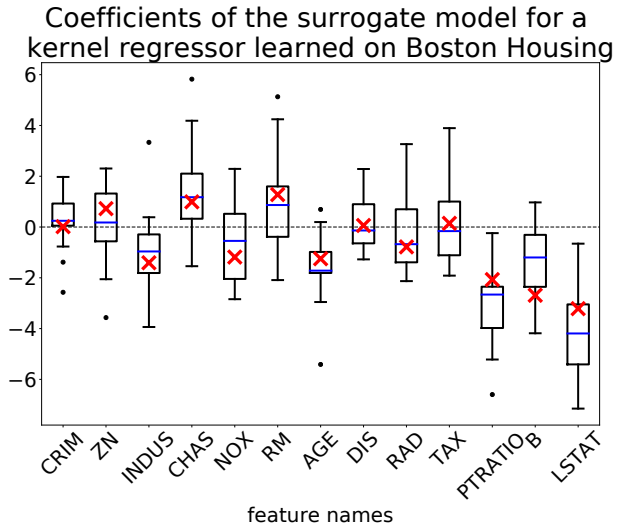

Figure 6: Values of the coefficients obtained by `TabularLIME` on each coordinate. We used the same settings as in Figure 5, but this time we train a *kernel ridge* regressor—a nonlinear function. For the ridge regression, we used the Gaussian kernel with scale parameter set to 5 and default regularization constant ($\alpha = 1$). We then estimated the partial derivatives of f at ξ and reported the corresponding β_js in red. For the chosen bandwidth (we took $\nu = 1$), the experiments seem to roughly agree with our theory.

accident) put θ_j to zero, therefore **forgetting** about feature j in the explanation, whereas it could play an important role in the prediction. To see why, we have to return to the definition of the θ_js: since $q_{j-} < q_{j+}$ by construction, to have $\theta_j = 0$ is possible only if

$$V_{\text{crit}} := \sigma^2 \frac{2\xi_j - q_{j-} - q_{j+}}{-2\mu_j + q_{j-} + q_{j+}} > 0\,, \qquad (4.1)$$

and ν^2 is set to V_{crit}. We demonstrate this switching-off phenomenon in Figure 7. An interesting take is that ν not only decides at which scale the explanation is made, but also the magnitude of the coefficients in the interpretable model, even for small changes of ν.

Error of the surrogate model. A simple consequence of Corollary 3.1 is that, unless some cancellation happens between in the term $f(\tilde{\mu}) - \sum_j \frac{a_j \theta_j}{\alpha_j}$, **the local error of the surrogate model is bounded away from zero**. For instance, as soon as $\tilde{\mu} \neq \mu$, it is the general situation. Therefore, the surrogate model produced by `TabularLIME` is not *accurate* in general. We show some experimental results in Figure 8.

Let us discuss briefly the limitations of Theorem 3.1.

Linearity of f. The linearity of f is a quite restrictive assumption, but we think that it is useful to consider for two reasons.

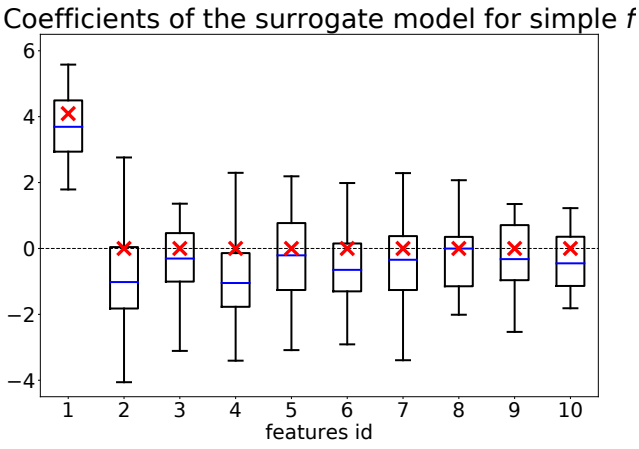

Figure 7: Values of the coefficients given by LIME. In this experiment, we took exactly the same setting as in Figure 4, but this time set the bandwidth to $\nu = 0.53$ instead of 1. In that case, the second feature is switched-off by `TabularLIME`. Note that it is not the case that ν is too small and that we are in a degenerated case: `TabularLIME` still puts a nonzero coefficient on the first coordinate.

First, the weighted nature of the procedure means that `TabularLIME` is not considering examples that are too far away from ξ with respect to the scaling parameter ν. Thus it is truly a *local* assumption on f, that could be replaced by a boundedness assumption on the Hessian of f in the neighborhood of ξ, at the price of more technicalities and assuming that ν is not too large. See, in particular, Lemma 11.3 in the Appendix, after which we discuss an extension of the proof when f is linear with a second degree perturbative term. We show in Figure 6 how our theoretical predictions behave for a non-linear function (a kernel ridge regressor).

Second, our main concern is to know whether `TabularLIME` operates correctly in a simple setting, and not to provide bounds for the most general f possible. Indeed, if we can already show imperfect behavior for `TabularLIME` when f is linear as seen earlier, our guess is that such behavior will only worsen for more complicated f.

Sampling strategy. In our derivation, we use the theoretical quantiles of the Gaussian distribution along each axis, and not prescribed quantiles. We believe that the proof could eventually be adapted, but that the result would loose in clarity.

5 Proof of Theorem 3.1

In this section, we explain how Theorem 3.1 is obtained. All formal statements and proofs are in the Appendix.

The main idea underlying the proof is to realize that $\widehat{\beta}$ is the solution of a weighted least squares problem.

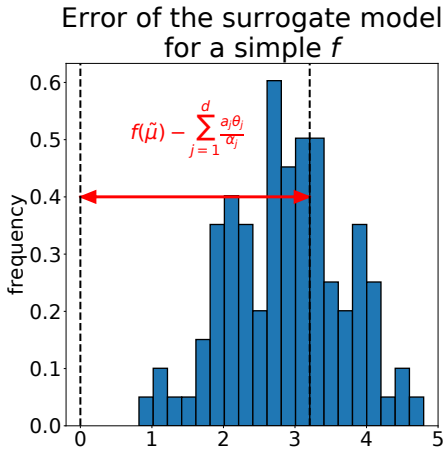

Figure 8: Histogram of the errors $\hat{f}(\xi) - f(\xi)$. The setting is the same as in Figure 4, but we repeated the experiment 100 times. The red double arrow marks the value given by Corollary 3.1 around which the local error concentrate. With high probability, the error of the surrogate model is bounded away from 0.

Denote by $\Pi \in \mathbb{R}^{n\times n}$ the diagonal matrix such that $\Pi_{ii} = \pi_i$ (the *weight matrix*), and set $f(x) \in \mathbb{R}^{d+1}$ the response vector. Then, taking the gradient of Eq. (5.1), one obtains the key equation

$$(Z^\top \Pi Z)\widehat{\beta} = Z^\top \Pi f(x)\,. \tag{5.1}$$

Let us define $\widehat{\Sigma} := \frac{1}{n} Z^\top \Pi Z$ and $\widehat{\Gamma} := \frac{1}{n} Z^\top \Pi f(x)$, as well as their population counterparts $\Sigma := \mathbb{E}[\widehat{\Sigma}]$ and $\Gamma := \mathbb{E}[\widehat{\Gamma}]$. Intuitively, if we can show that $\widehat{\Sigma}$ and $\widehat{\Gamma}$ are close to Σ and Γ, assuming that Σ is invertible, then we can show that $\widehat{\beta}$ is close to $\beta := \Sigma^{-1}\Gamma$.

The main difficulties in the proof come from the **non-linear** nature of the new features z_i, introducing tractable but challenging integrals. Fortunately, the Gaussian sampling of LIME allows us to overcome these challenges (at the price of heavy computations).

Covariance matrix. The first part of our analysis is thus concerned with the study of the empirical covariance matrix $\widehat{\Sigma}$. Perhaps surprisingly, it is possible to compute the population version of $\widehat{\Sigma}$:

$$\Sigma = C_d \begin{pmatrix} 1 & \alpha_1 & \cdots & \alpha_d \\ \alpha_1 & \alpha_1 & & \alpha_i\alpha_j \\ \vdots & & \ddots & \\ \alpha_d & \alpha_i\alpha_j & & \alpha_d \end{pmatrix},$$

where the α_js were defined in Section 3, and C_d is a scaling constant that does not appear in the final result (see Lemma 8.1).

Since the α_js are always distinct from 0 and 1, the special structure of Σ makes it possible to invert it in closed-form. We show in Lemma 8.2 that

$$C_d^{-1} \begin{pmatrix} 1 + \sum_{j=1}^d \frac{\alpha_j}{1-\alpha_j} & \frac{-1}{1-\alpha_1} & \cdots & \frac{-1}{1-\alpha_d} \\ \frac{-1}{1-\alpha_1} & \frac{1}{\alpha_1(1-\alpha_1)} & & 0 \\ \vdots & & \ddots & \\ \frac{-1}{1-\alpha_d} & 0 & & \frac{1}{\alpha_d(1-\alpha_d)} \end{pmatrix}.$$

We then achieve control of $\left\|\widehat{\Sigma}^{-1} - \Sigma^{-1}\right\|_{\text{op}}$ *via* standard concentration inequalities, since the new samples are Gaussian and the binary features are *bounded* (see Proposition 8.1).

Right-hand side of Eq. (5.1). Again, despite the non-linear nature of the new features, it is possible to compute the expected version of $\widehat{\Gamma}$ in our setting. In this case, we show in Lemma 9.1 that

$$\Gamma = C_d \begin{pmatrix} f(\tilde{\mu}) \\ \alpha_1 f(\tilde{\mu}) - a_1\theta_1 \\ \vdots \\ \alpha_d f(\tilde{\mu}) - a_d\theta_d \end{pmatrix},$$

where the θ_js were defined in Section 3. They play an analogous role to the α_js but, as noted before, they are signed quantities. As with the analysis of the covariance matrix, since the weights and the new features are bounded, it is possible to show a concentration result for $\widehat{\Gamma}$ (see Lemma 9.3).

Concluding the proof. We can now conclude, first upper bounding $\left\|\widehat{\beta} - \Sigma^{-1}\Gamma\right\|$ by

$$\left\|\widehat{\Sigma}^{-1}\right\|_{\text{op}} \left\|\widehat{\Gamma} - \Gamma\right\| + \left\|\widehat{\Sigma}^{-1} - \Sigma^{-1}\right\|_{\text{op}} \|\Gamma\|\,,$$

and then controlling each of these terms using the previous concentration results. The expression of β is simply obtained by multiplying Σ^{-1} and Γ.

6 Conclusion and future directions

In this paper we provide the first theoretical analysis of LIME, with some good news (LIME discovers interesting features) and bad news (LIME might forget some important features and the surrogate model is not faithful). All our theoretical results are verified by simulations.

For future work, we would like to complement these results in various directions: Our main goal is to extend the current proof to any function by replacing f by its Taylor expansion at ξ. On a more technical side, we would like to extend our proof to other distance functions (*e.g.*, distances between the z_is and ξ, which is the default setting of LIME), to non-isotropic sampling of the x_is (that is, σ not constant across the dimensions), and to ridge regression.

Acknowledgements

The authors would like to thank Christophe Biernacki for getting them interested in the topic, as well as Leena Chennuru Vankadara for her careful proofreading. This work has been supported by the German Research Foundation through the Institutional Strategy of the University of Tübingen (DFG, ZUK 63), the Cluster of Excellence "Machine Learning—New Perspectives for Science" (EXC 2064/1 number 390727645), and the BMBF Tuebingen AI Center (FKZ: 01IS18039A).

Bibliography

D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(6):1803–1831, 2010.

S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press, 2013.

R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5):93, 2019.

D. Harrison Jr. and D. L. Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management*, 5(1):81–102, 1978.

S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *NeurIPS*, 2017.

D. P. O'Leary and G. W. Stewart. Computing the eigenvalues and eigenvectors of arrowhead matrices. *Journal of Computational Physics*, 90:497–505, 1996.

X. Ren and J. Malik. Learning a classification model for segmentation. In *ICCV*, 2003.

M. T. Ribeiro, S. Singh, and C. Guestrin. Why should I trust you? explaining the predictions of any classifier. In *SIGKDD*, 2016.

L. S. Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *ICML*, 2017.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

A. W. Van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 2000.

M. J. Wainwright. *High-dimensional statistics: a non-asymptotic viewpoint*. Cambridge University Press, 2019.

H. Weyl. Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung). *Mathematische Annalen*, 71(4):441–479, 1912.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833, 2014.