Using Stratified Sampling to Improve LIME Image Explanations

Muhammad Rashid[1], **Elvio G. Amparore**[1], **Enrico Ferrari**[2], **Damiano Verda**[2]

[1]University of Torino, Computer Science Department, C.so Svizzera 185, 10149 Torino, Italy
[2]Rulex Innovation Labs, Via Felice Romani 9, 16122 Genova, Italy
{muhammad.rashid, elviogilberto.amparore}@unito.it, {enrico.ferrari, damiano.verda}@rulex.ai

Abstract

We investigate the use of a stratified sampling approach for LIME Image, a popular model-agnostic explainable AI method for computer vision tasks, in order to reduce the artifacts generated by typical Monte Carlo sampling. Such artifacts are due to the undersampling of the dependent variable in the synthetic neighborhood around the image being explained, which may result in inadequate explanations due to the impossibility of fitting a linear regressor on the sampled data. We then highlight a connection with the Shapley theory, where similar arguments about undersampling and sample relevance were suggested in the past. We derive all the formulas and adjustment factors required for an unbiased stratified sampling estimator. Experiments show the efficacy of the proposed approach.

Introduction

The efficacy of explainable AI techniques for computer vision tasks has seen several important advancements in the recent years. Several methods to interpret model predictions have emerged, as surveyed for instance by (Liang et al. 2021) or (Guidotti et al. 2018). In this paper we inspect the sampling strategy of one of these methods known as *LIME Image* (Ribeiro, Singh, and Guestrin 2016), which is a *model-agnostic* method (i.e. it is not tied to a particular type of black box model being explained) that produces *feature attributions* as explanations. As the name suggests, LIME Image is a method specialized for image classification tasks, and the "feature attribution" are importance scores assigned to regions of an input image measuring how much each region contributes to the model classification.

Feature attributions are the regression coefficients that solve a weighted least squares problem on a sampled population denoted as *synthetic neighborhood*. Since the sampling process is inherently stochastic, the synthetic neighborhood may be inadequate for LIME Image to fit the regressor, resulting in slow convergence (Visani et al. 2022) or instability (Sevillano-García et al. 2022). Sometimes, the explanation produced by LIME Image fails to identify any relevant region, resulting in regression coefficients with very small and almost uniform values (i.e. with low *variation*, as we shall see). We review the LIME Image process, focusing on the limitation of using a Monte Carlo sampling for the synthetic neighborhood generation.

Paper Contributions. In this paper we:

- investigate the distribution of the dependent variable in the sampled synthetic neighborhood of LIME Image, identifying in the undersampling a cause that results in inadequate explanations;
- delve into the causes of the synthetic neighborhood inadequacy, recognizing a link with the Shapley theory;
- reformulate the synthetic neighborhood generation using an *unbiased stratified sampling* strategy;
- provide empirical proofs of the advantage of using stratified sampling for LIME Image on a popular dataset.

Previous Work

A relevant theoretical study of LIME Image is (Garreau and Mardaoui 2021), which we partially summarize in the *Preliminaries* section for the sake of self-containment, that also focuses on connections with *integrated gradients*. Discretization of the synthetic neighborhood for tabular data has been studied in (Garreau and Luxburg 2020), and for text data by (Mardaoui and Garreau 2021). However, the setting for image data is significantly different, since the sample space is Boolean and not continuous. Sampling strategies received more attention in the context of the Shapley theory (Lundberg and Lee 2017), as in (Mitchell et al. 2022). We recast some of the intuitions of these previous works in the context of LIME, particularly from the *multilinear extensions* (Owen 1972).

Several alternative sampling strategies for LIME have been studied. A clique-based sampling was considered in (Shi, Du, and Fan 2020). Moreover, sampling variance has been considered in several articles like (Zhang et al. 2019) or in (Shankaranarayana and Runje 2019), where standard deviations of Ridge coefficients are compared. A complementary study about region flipping analysis in LIME explanations is (Ng, Abuwala, and Lim 2022), which could also be used to improve the approach proposed in this paper. To the best of our knowledge, we are not aware of a consistent framework that adds unbiased stratified sampling to LIME.

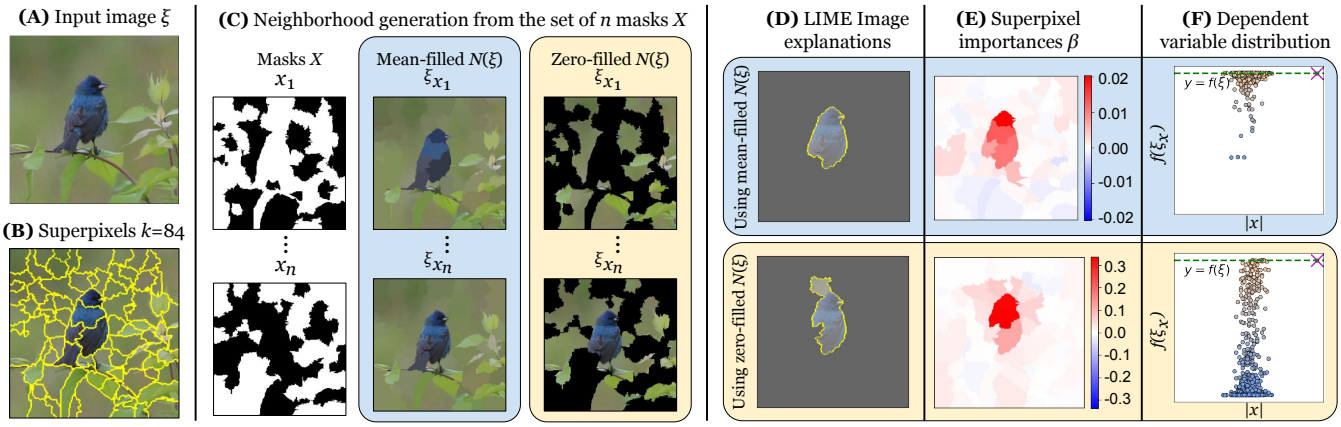


Figure 1: LIME Image workflow.

Preliminaries

We briefly review how LIME works for image inputs, in order to explain our changes and their effects. Fig. 1 depicts the LIME Image workflow steps, and will be used throughout this section to provide examples. Consider the domain of RGB images of size $h \times w$, denoted as $\mathcal{I} \in [0-255]^{h \times w \times 3}$. Let $f : \mathcal{I} \to \mathbb{R}$ be a black-box regression model function that provides a prediction score given an input image[1], and let $\xi \in \mathcal{I}$ be the sample image being explained. The main purpose of LIME is to generate a *linear* model g that locally approximate the explained black-box model f in the neighborhood of an input sample ξ.

LIME explanations are not build directly on the image $\mathcal{I}$, but on a smaller domain denoted as the *interpretable representation*. This domain is obtained by divided the input image into k *superpixels* (also called *segments*, *regions* or *patches*) using an algorithm like *quick shift* (Vedaldi and Soatto 2008). A superpixel is a contiguous region of pixels of ξ that share some kind of similarity, and such that the k superpixels form a partition of the pixels of ξ. Fig. 1A shows an example of an image taken from (Addison Howard 2018). 1B shows its segmentation obtained from the *quick shift* algorithm[2], resulting in $k = 84$ superpixels. The model being used for the classification is ResNet50 (He et al. 2016), pretrained for the ImageNet task. The image in Fig. 1A is correctly classified as *indigo_bunting* with probability 99.49%.

The approach of LIME Image is based on the concept of *superpixel masking*. Let $x \in \{0,1\}^k$ be a binary vector (mask) representing the presence (value 1) or the absence (value 0) of each of the k superpixels. Giving a mask x, a *perturbed input image* ξ_x is obtained by preserving the pixels of each superpixel i having $x[i] = 1$, and replacing every other pixel whose superpixel i has $x[i] = 0$. Replacement can be done in several ways. By default pixels of a masked superpixel i are replaced by the mean color of that superpixel (*mean-filled*). Alternatively, they can be replaced with a fixed color value, like black (*zero-filled*). We use notation $x' = x[i \leftarrow v]$ to denote a new mask x' obtained from a mask x by replacing the value for superpixel i with v. Moreover, let $|x|$ be the number of preserved superpixels, i.e. those having $x[i] = 1$.

In LIME Image, the individual values of a mask vector x are sampled using an unbiased Monte Carlo strategy, i.e.

$$x[i] \sim B(0.5), \qquad 1 \leq i \leq k \tag{1}$$

where $B(p)$ is a Bernoulli-distributed random variable having probability p=0.5. A *set of masks* X with n samples is made by randomly sampling n instances of (1) for the same input image ξ having k superpixels. A *synthetic neighborhood* $N(\xi) = \{\xi_x \mid x \in X\}$ with n samples is made by perturbing the input image ξ using n randomly sampled masks. A depiction of the set of n masks is shown in Fig. 1C: randomly sampled masks x_i are used to generate perturbed input images ξ_{x_i}, using two replacement strategies.

All the perturbed samples $N(\xi)$ can be classified by the black-box model f, resulting in the *dependent variables*

$$Y = \{f(\xi_x) \mid \xi_x \in N(\xi)\} \tag{2}$$

A *distance function* is adopted, in order to weight the perturbed samples differently. The intuition followed by LIME is that samples closer to ξ should weight more.
Given a mask x, the weight w_x is

$$w_x = \exp\left(\frac{-D(x)^2}{\sigma^2}\right) \tag{3}$$

where D is the cosine similarity score between x and $\vec{1}$ (the vector of ones, i.e. the mask where everything is preserved), while $\sigma = 0.25$ (by default) is the *kernel width*. See (Garreau and Luxburg 2020) for an analysis on the role of Eq. (3) and of σ. In this paper we will use the default value, as the focus is in the sampling methodology. Let $W = \{w_x \mid x \in X\}$.

Having the matrices of the set of masks $X \in \{0,1\}^{n \times k}$, the weights $W \in \mathbb{R}^{n \times 1}$ and the dependent variables $Y \in \mathbb{R}^{n \times 1}$ for all the observed samples in the synthetic neighborhood $N(\xi)$, then Y can be written as the response variable of the *linear regression model*. LIME adopts a *simple linear*

[1]We consider only the case of a binary class prediction, as the multi-class prediction is usually treated as several one-vs-rest binary class prediction problems.

[2]Using: $kernel\_size = 4$, $max\_dist = 7$, $ratio = 0.2$.

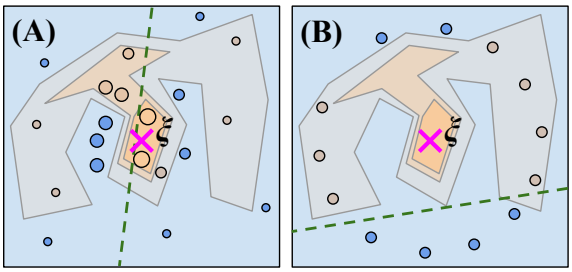


Figure 2: How LIME is supposed to work (A), and how it actually works (B) using Monte Carlo sampling for a large enough k.

homoscedastic model (DuMouchel and Duncan 1983) for its regression coefficients, which is

$$Y = X \cdot \beta + \epsilon \tag{4}$$

where the vector β is the weighted least squares estimator of the regression coefficients of Y on X weighted by W.

To simplify our analysis, we will consider no regularization factors (default for LIME Image is ridge regression with L^2 regularization), similarly to (Garreau and Luxburg 2020). This simplification does not affect significantly the main observations of this paper, which is focused on the sampling strategy. The coefficients β results from

$$\beta = (X^\mathsf{T} W X)^{-1} X^\mathsf{T} W Y \tag{5}$$

which solves Eq. (4). A linear function $g(x)$ with coefficients β is a linear regressor that locally approximates the initial black-box model f.

Interpretation of LIME. The k coefficients of β can be interpreted as *feature importances* (or *feature attributions*) of each of the k superpixels of the input image ξ. In that sense, the k superpixels form the set of *interpretable features* of the input image, over which the explanation is built.

There are two levels of interpretation of β. By default LIME Image suggests to select only the superpixels with the highest value (Fig. 1D), resulting in an sub-region in the image (the get_image_and_mask method). The number of selected superpixels is decided by the user: LIME does not provide an heuristic for this task. Alternatively, the coefficients can be visualized as an *heatmap*, identifying the contribution of each superpixel to the classification (Fig. 1E). The color intensity represents the value, with white representing the zero. Coefficients with higher absolute values means that the corresponding superpixel is more important in the classification outcome $f(\xi)$. The scale of the coefficients can vary (in Fig. 1E the same scale is used for both heatmaps) and it is known to not be particularly relevant (Garreau and Luxburg 2020, pag. 6) (only the ratios among the coefficients is).

Finally, it is relevant to inspect the distribution of the Y values in the neighborhood (i.e. the values of $f(\xi_x)$) with respect to the count $|x|$ of masked superpixels (Fig. 1F). This plot shows if the Y values are sampled across the entire distribution (top to bottom), or if there are clear unbalances. In

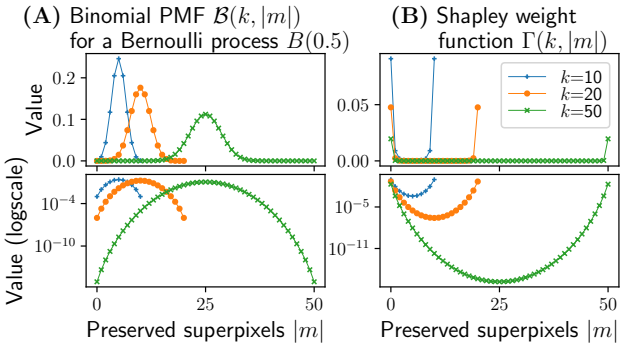


Figure 3: Binomial (A) and Shapley weight (B) distributions for $k = 10, 20$ and 50.

Fig. 1F, the distribution for the zero-filled case has a good balance, since there are values obtained from the black box model f covering the whole spectrum of values, while in the plot for the mean-filled case the balance is problematic, having most Y values concentrated in the top. As we shall see in the next section, imbalances in this distribution results in poor explanations being generated by LIME Image.

Limitations of LIME Image Sampling

While there has been a number of successful applications of LIME (Bodria et al. 2023), the explanation process largely depends on several factors. One such factors is the sampling process, which is stochastic and inherently uncertain. The use of a Monte Carlo strategy in Eq. (1) to sample the interpretable feature space when it is made by more than a few dozen of superpixels has important consequences.

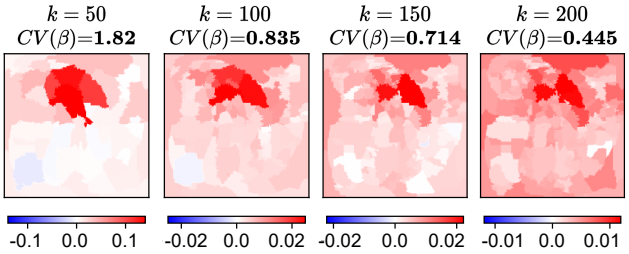
Under-Representation of the Neighborhood. The intuition behind LIME is depicted in Fig. 2A, which is inspired by the one found in (Ribeiro, Singh, and Guestrin 2016, Fig. 3). The explained sample ξ (represented as a cross) is surrounded by its synthetic neighborhood $N(\xi)$ (represented as dots), whose classifications are obtained by the black box model f and weighted by their proximity to ξ (size of dots). A linear regressor (the green dashed line) is fit on these points weighted by their distance to ξ, and in principle it should be locally faithful to $f(\xi)$. LIME Image however works like that only when the number of superpixels is very small. Since masks are obtained from Eq. (1) having a fixed Bernoulli coefficient of 0.5, the probability of selecting a mask x having a given number of preserved superpixels $|x|$ follows the binomial distribution $\mathcal{B}(k, |x|)$ with probability mass function $\binom{k}{|x|} p^{|x|}(1-p)^{k-|x|}$.

Fig. 3A shows the probability mass function for a few k values, being k the number of superpixels. This PMF is of course not uniform, and the probability of randomly sample points at the extremes drops rapidly. There is no indication of how many superpixels LIME Image can manage, but both the default parameters and practical experience (Vermeire et al. 2022) shows that an image needs to be split into tens or even a few hundreds of superpixels, in order to have enough patches to correctly identify object borders.

(A) Input image ξ
Class: *hyena*
Probability: 99.46%
num_samples n = 1000
Using mean-filled $N(\xi)$

k	*max_dist*
50	8.691
100	4.956
150	4.092
200	3.632

(B) Feature importances β for four segmentations.



(C) Dependent variable distributions.

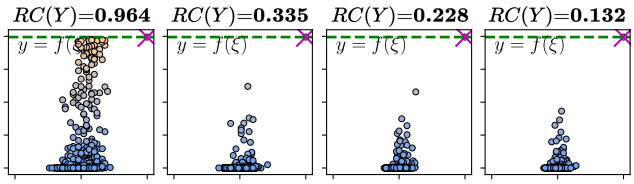


Figure 4: Dependent variable undersampling (low $RC(Y)$) results in confused explanations (low $CV(\beta)$).

In that case, samples will distribute around ξ forming a sort of hypersphere, as illustrated in Fig. 2B, where almost no sample is really close to ξ, since the probability of the binomial distribution concentrates around samples having $\sim$50% of the superpixels masked. In that way, the local behaviour (i.e. samples with $|x|$ close to k) is under-represented in the neighborhood.

Dependent Variables Distribution. As seen in Fig. 3A, by increasing the superpixels k the probability of getting samples from the tails of the distribution is practically reduced to 0. This effect depends on both the model and the input image: if selecting randomly about 50% of the superpixels still allows the model to produce a "reasonable" distribution of the dependent variable Y, a linear regressor can be fit and an explanation can be produced. If however the Y distribution is flattened, no reasonable explanation can be produced, as the linear regressor will be fit on almost uniform values.

Fig. 4 shows an example of this behaviour. The input image **(A)** is correctly classified by the model as *hyena* with high probability. Feature importance vectors β and the distribution of the dependent variables Y (versus the number of masked superpixels $|x|$) are shown in **(B)** and **(C)**, respectively, for four different segmentations ($k = 50, 100, 150$ and 200 superpixels, respectively). All values (heatmaps, $CV(\beta)$, $RC(Y)$) are averages of 10 computations, to reduce randomicity in the reported results. With $k = 50$ segments (left), the Y distribution has enough variability to obtain a vector β that highlights which segments are more important. Increasing the number of superpixels reduces such variability in the Y distribution, resulting in explanations that are more and more "confused". On these distributions it is of course harder to fit a linear regressor that is truthful to the explanation. Intuitively, it is like Fig. 2B where the hypersphere is almost entirely far away from ξ. In that case, the explanation produced by LIME Image will be progressively more meaningless.

In these problematic cases the values of the β vector also drops to very small numbers (scale is reported below each heatmap in **(B)**), and variability across the feature importances decreases. To quantitatively measure such form of "confusion", we employ the standard *coefficient of variation*, defined as

$$CV(\beta) = \frac{\sigma_\beta}{\mu_\beta} \tag{6}$$

where σ_β and μ_β are the standard deviation and the mean of β, respectively. Ideally, a good $CV(\beta)$ should not be close to zero (which would mean that all superpixels have almost the same value, and no clear sub-region in the image is identified). The $CV(\beta)$ values for the example in Fig. 4 are reported in the **(B)** row.

We also want to quantify the (approximate) *range coverage* of the Y values in the synthetic neighborhood. Theoretically this range is $[0, f(\xi)]$, but of course it can have under- or over-shoots due to the nature of the classification model. To do so, we measure the proportion of that range that is contained in the $1\% - 99\%$ interquantile range (IQR) of the Y distribution, using

$$RC(Y) = \frac{IQR_{1-99}(Y)}{f(\xi)} \tag{7}$$

Low values of $RC(Y)$ indicate that the sampled Y distribution is squashed into a small range of values, not covering the full $[0, f(\xi)]$ spectrum (like in Fig. 4C/right). Ideally $RC(Y)$ should be far from zero to have a good coverage of the probability range $[0, f(\xi)]$ by Y.

Sample Relevance. In the recent years, the Shapley theory (Lundberg and Lee 2017) has received a lot of attention in the context of model-agnostic explainability, due to its flexibility and its axiomatic formulation (Rozemberczki et al. 2022). While LIME does not have a corresponding axiomatic definition, we can still learn some insights from how Shapley values are defined over a weight sample space.

The Shapley value for a superpixel i, that can be interpreted as an *importance* score, is defined by

$$\phi_i = \sum_{x \in X^{[\![i]\!]}} \Gamma(k-1, |x|)\big(f(\xi_{x[i\leftarrow 1]}) - f(\xi_x)\big) \tag{8}$$

with $X^{[\![i]\!]}$ being the set of all masks x having $x[i] = 0$, and with the *Shapley importance* function (Monderer and Samet 2002, p. 6)

$$\Gamma(k, |x|) = \frac{1}{(k+1)\binom{k}{|x|}} \tag{9}$$

Fig. 3B shows the Shapley importance function for a few k values. Higher values of $\Gamma(k, |x|)$ for a mask x means that samples having that mask will weight more in the final value of ϕ_i. Comparing Fig. 3A and B clearly shows that LIME

Image samples the majority of the masks among those having the least importance (in the Shapley sense). In fact when $p = 0.5$ it holds that

$$\mathcal{B}(k, |x|) \cdot \Gamma(k, |x|) = \frac{\binom{k}{|x|} p^{|x|}(1-p)^{k-|x|}}{(k+1)\binom{k}{|x|}} = \frac{0.5^k}{k+1}$$

i.e. the Shapley importance is the reciprocal (times a constant) of the binomial distribution $B(0.5)$ used by LIME. This is an informative detail of the Shapley theory, which motivates the proposed sampling theory.

Interestingly, Shapley value computation is not typically performed as a Monte Carlo sampling, but adopts other strategies to generate the samples (Okhrati and Lipani 2021; Mitchell et al. 2022). For instance, in (Owen 1972) Eq. (8) is rewritten as

$$\phi_i = \int_0^1 \left(\sum_{x \in X_q^{[\![i]\!]}} \frac{1}{|X_q^{[\![i]\!]}|} \big(f(\xi_{x[i\leftarrow 1]}) - f(\xi_x) \big) \right) \mathrm{d}q \quad (10)$$

with $X_q^{[\![i]\!]}$ being a random subset of masks x, having $x[i] = 0$ and, for all $j \neq i$, $x[j] \sim B(q)$ with $B(q)$ a Bernoulli-distributed random variable having probability q. Such strategy allows to get samples across the entire spectrum of $|x|$ values. In the rest of the paper we shall discuss a strategy for LIME Image where x values are not sampled from $B(0.5)$ as in Eq. (1) but from a modified version of Eq. (10).

Proposed Methodology

We describe a methodology based on stratified sampling of the X values, where each stratum has a uniform probability of being selected and represented in the samples of X. This oversamples the "rare" samples at the tail of the Y distribution, improving the samples over which the linear regressor is fit. However, this sampling could result in a form of *bias*. To avoid that, an adjustment factor is introduced to counterbalance the oversampled data points.

Let $\mathcal{X}$ denote the complete population of mask samples, having 2^k elements, and let $\mathcal{Y}$ be the dependent variable of $\mathcal{X}$. Consider a stratified partitioning. Let $\mathcal{X}^{(i)}$ be the set of all possible masks having $|x| = i$, i.e. for which exactly i superpixel are preserved.. Clearly, $\mathcal{X}^{(0)} \cdots \mathcal{X}^{(k)}$ forms a partitioning of all possible masks, and

$$\{0,1\}^k = \bigcup_{i=0}^{k} \mathcal{X}^{(i)}$$

since any possible mask x appears in one (and only one) set $\mathcal{X}^{(|x|)}$. Moreover $\mathcal{X}^{(0)}{=}\{\vec{\mathbf{0}}\}$ and $\mathcal{X}^{(k)}{=}\{\vec{\mathbf{1}}\}$ (masks for the explained input sample with everything/nothing perturbed, resp.). Each stratum $\mathcal{X}^{(i)}$ does not have a uniform number of samples, but its size is known a-priori since they follow the binomial distribution, i.e.

$$|\mathcal{X}^{(i)}| = \binom{k}{i}, \qquad 0 \leq i \leq k \quad (11)$$

In an unbiased Monte Carlo sampling model, as Eq. (1), the probability of selecting a sample x in a from stratum $\mathcal{X}^{(i)}$, with $i = |x|$, is therefore proportional to that stratum probability in the overall population $\mathcal{X}$, i.e.

$$Prob\big\{x \in \mathcal{X}^{(i)} \mid x \in X\big\} = \frac{|\mathcal{X}^{(i)}|}{\sum_{j=0}^{k} |\mathcal{X}^{(j)}|} = \frac{\binom{k}{i}}{2^k}$$

Let $\widehat{X}$ be an oversampled population, where the probability of taking samples from any of the $k+1$ strata is uniform, and does not depend on the stratum size, i.e.

$$Prob\big\{x \in \mathcal{X}^{(i)} \mid x \in \widehat{X}\big\} = \frac{1}{k+1}$$

Let $\widehat{Y}$ be the corresponding dependent variables for $\widehat{X}$. We can derive an *adjustment factor* for the $\widehat{X}$ samples to correct the bias introduced by the oversampling, which results for an arbitrary sample x in stratum $\mathcal{X}^{(i)}$ as

$$adj(i) = \frac{Prob\big\{x \in \mathcal{X}^{(i)} \mid x \in X\big\}}{Prob\big\{x \in \mathcal{X}^{(i)} \mid x \in \widehat{X}\big\}} = \frac{(k+1)\binom{k}{i}}{2^k} \quad (12)$$

Weighted regression with the oversampled set $\widehat{X}$ can be obtained by inserting the adjustment factor as a multiplicative term in the existing weight equation of LIME. Let $\widehat{w}_{\widehat{x}}$ be the weight of sample $\widehat{x} \in \widehat{X}$ obtained from Eq. (3) multiplied by $adj(|\widehat{x}|)$, and let $\widehat{W} = \{\widehat{w}_{\widehat{x}} \mid \widehat{x} \in \widehat{X}\}$ be the set of weights for the set $\widehat{X}$. Then let

$$\widehat{\beta} = (\widehat{X}^{\mathsf{T}} \widehat{W} \widehat{X})^{-1} \widehat{X}^{\mathsf{T}} \widehat{W} \widehat{Y} \quad (13)$$

be the weighted least square estimator of the regression coefficients of $\widehat{Y}$ on $\widehat{X}$ that takes into account the strata density of the oversampled set $\widehat{X}$.

The Mixture Model. The linear homoscedastic regression model of Eq. (4) adopted by LIME may not be particularly accurate when strata at the tails are severely undersampled, and these strata are significantly different from the mean. In that case, β is not globally unique across the sampled population, but varies by stratum

$$\widehat{Y}^{(i)} = \widehat{X}^{(i)} \cdot \widehat{\beta}^{(i)} + \widehat{\epsilon}^{(i)} \quad (14)$$

Intuitively, the $\widehat{\beta}^{(i)}$ vectors represents the feature importance for stratum i, which is at uniform "distance" from the input sample ξ. The closer i is to k, the closer ξ_x is to ξ.

Impact of Stratified Sampling in LIME Image. The impact of using a weighted regression from a stratified sampling schema may not be negligible. We simplify the analysis considering two cases.

Case (A): The mean and variance of $\widehat{\beta}^{(i)}$ are independent of the strata (i.e. the population structure is *homoscedastic*). Then it is easy to see that $\mathbb{E}[\beta] \approx \mathbb{E}[\widehat{\beta}^{(i)}]$, for any i. In that case, a weighted regression model of Eq. (13) is not needed, and the model computed by LIME using Monte Carlo sampling will not have issues due to the undersampling of the tails. In that case, the stratified sampling will converge to the same values, regardless of the strata ratios in the synthetic neighborhood.

Algorithm 1: Neighborhood sampling strategies

function *MonteCarloSampling*(n, k)
1 $X \leftarrow n \times k$ matrix ;
2 **for** i *between 1 and* n **do**
3 **for** j *between 1 and* k **do**
4 $X[i, j] \leftarrow B(0.5)$

function *StratifiedSampling*(n, k)
1 $X \leftarrow n \times k$ matrix ;
2 **for** i *between 1 and* n **do**
3 $q \leftarrow Uniform(0, 1)$;
4 **for** j *between 1 and* k **do**
5 $X[i, j] \leftarrow B(q)$;
6 $adj[i] \leftarrow (k + 1) \cdot \frac{1}{2^k} \cdot \binom{k}{|X[i]|}$

Case (B): The mean and variance of $\widehat{\beta}^{(i)}$ varies by stratum. In that case, the bias introduced by the Monte Carlo sampling scheme will not allow to consider the systematic differences in the stratum, and a weighted regression or a mixed model built on a stratified sampling strategy are highly advisable (DuMouchel and Duncan 1983).
In a certain sense Case (B) is even worse, because the undersampling of the neighborhood of ξ breaks the logic of building models that are locally faithful to the black box model f in the neighborhood of the explained sample, since the local neighborhood (close to ξ) that is really representing the local behaviour is missing/undersampled.

Algorithm 1 outlines two sampling methods: the original Monte Carlo sampling used by LIME Image, and the introduced stratified sampling technique. The *MonteCarloSampling* function computes the data matrix X (from Eq. 1) with replacement. Function *StratifiedSampling* is one possible way of generating a stratified population, similarly to Eq. (10). For every sample i, a single coefficient q is randomly drawn from a uniform distribution ranging between 0 and 1. The individual values of the i-th mask vector are then sampled from a Bernoulli random variable $B(q)$ with probability q. This will obtain a sample $X[i]$ in stratum $\widehat{X}^{(|X[i]|)}$, where strata have now equal probability of being selected. The adjustment factor $adj[i]$ for sample i is also computed.

Other strategies could also be employed (Rao 1977). An interesting approach suggested in (Konijn 1962) for computing the coefficients would be to fit one linear regressor for every strata and then form a *mixed model* with the coefficients' averages. This approach however requires more changes in the LIME code, thus we have favored the approach of Algorithm 1 which is more straightforward.

Experimental Evaluation

We perform experiments to compare the proposed methodology with the original Monte Carlo setup of LIME, in order to test whether the generated distributions of $\widehat{Y}$ have a better sampling, resulting in feature attribution vectors $\widehat{\beta}$ that are less confused.

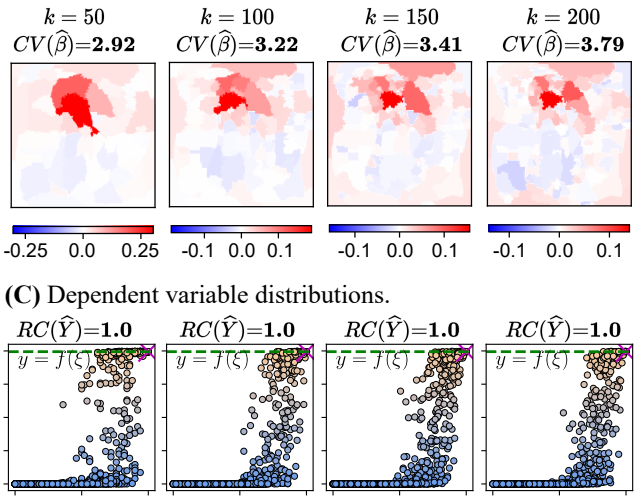


Figure 5: Four explanations $\widehat{\beta}$ of the same image of Fig. 4 using stratified sampling (each is an average of 10 runs).

We start by revisiting the *hyena* example of Fig. 4 but recomputed using the *StratifiedSampling* algorithm. The results are reported in Fig. 5. The first thing to observe is that the dependent variable distribution has now samples for several different classification scores, which allows the linear regressor to be fit against a synthetic neighborhood with better variation than in the standard Monte Carlo setup of Fig. 4B. The heatmap of the explanations also reflect this improvement: feature attribution values now have a much better coefficients of variation, resulting is some superpixels receiving high importance, and other receiving almost zero importance. Moreover, the explanation remains reasonably consistent, identifying the same "spot" in the image even when the set of superpixels changes. Moreover, Fig. 5C shows that the distribution of the dependent variable (the y-axis) across the strata (the $|x|$ value on the x-axis) is far from being homoscedastic. This further reinforces the need for stratified sampling in the process.

To better quantify the effect, we took the first 150 images of the *ImageNet Object Localization dataset* (Addison Howard 2018). For each image we performed a dichotomic search on the *max_dist* hyperparameter to find a configuration of *quick shift* that results in a number of superpixels k equal to 50, 100, 150 and 200. For each range, we run 10 times LIME Image with both the Monte Carlo and the stratified sampling using n=1000 samples, and record both the average range coverage RC of the Y ($\widehat{Y}$ resp.) distributions and the CV of the feature attribution vectors β ($\widehat{\beta}$ resp.). The first two rows of plots in Fig. 6 show the results obtained from Monte Carlo (above) and stratified sampling (below). Each plot has 150 dots, one for each image in the dataset for a fixed k. Each dot has the CV on the y-axis, and the range coverage RC on the x-axis. It is very clear that the stratified sampling approach ensures that the range of $\widehat{Y}$ distribution range is well covered w.r.t.

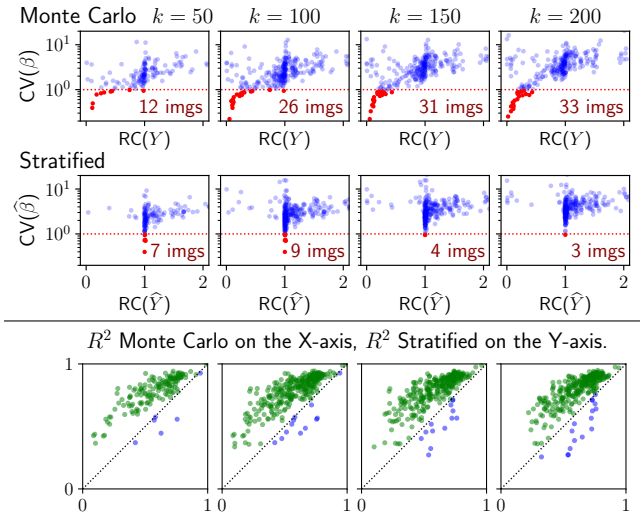


Figure 6: CV vs RC and R^2 comparisons, for 150 images.

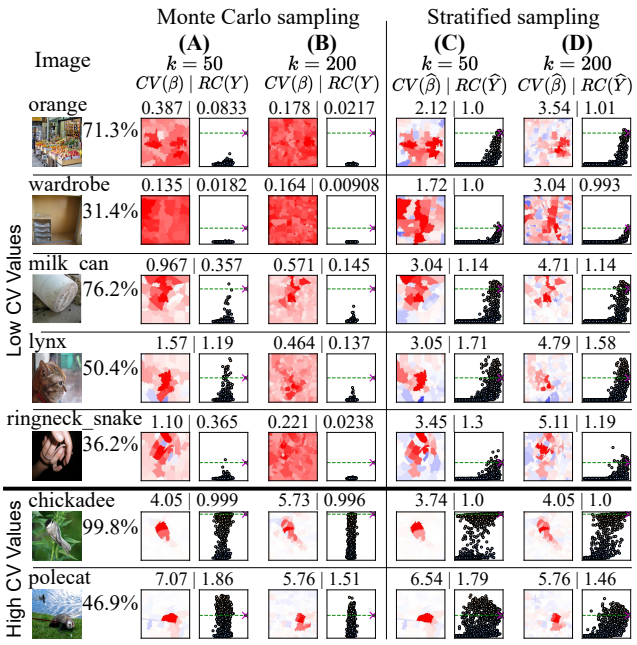


Figure 7: Examples of LIME Image explanations in the lower-left tail of Fig. 6, with heatmaps, CV and RC values.

the Y distribution. At the same time, the Monte Carlo approach produces, for some images, explanations with very poor variation in the coefficients, and this is clearly linked with the low range coverage. Explanations with an average CV below one are highlighted. The third row in Fig. 6 reports the comparison of the average R^2 coefficients for the Stratified (on the y axis) and for the Monte Carlo (on the x axis), showing that, on average, the $\widehat{Y}$ distribution better explains the X distribution than Y.

We report some of these images with low CV values in Fig. 7 (first five rows). Columns A and B show the Monte Carlo sampling, C and D the Stratified sampling. We consider the cases with k=50 (columns A and C) and k=200 (columns B and D). For each explanation we show the heatmap and the Y ($\widehat{Y}$ resp.) distribution, together with the CV and RC values. Column B clearly shows the problem: the Monte Carlo sampled distributions are very poor, with all Y almost close to 0. This results in feature attribution vectors β that are almost uniform, which do not identify any relevant sub-region of the explained images. This detrimental effect is greatly reduced by the stratified sampling approach, which remains capable of identifying a sub-region of the image that is deemed to be responsible for the classification. When the sampled distribution is sufficient, both the Monte Carlo and the Stratified sampling approaches converge to similar explanations (last 2 rows of Fig. 7).

Conclusions

We have provided a reformulation of the sampling strategy of LIME Images showing its critical role in cases where the simple linear homoscedastic model for regression is not true, i.e. when the Y value are undersampled by a Monte Carlo strategy. This happens when the black-box model f (almost always) returns low classification scores when about $\sim 50\%$ of the explained image ξ is masked, resulting in flat Y distributions with very low range coverage, for which the coefficient β of a linear regression model will be close-to constant (i.e. with low variation). We considered image data, using the popular ImageNet dataset for the experiments. Of course the strategy could be of interest for other kind of data, even if some adjustments are probably needed (since the interpretable feature space for images is over the booleans, unlike for other data types). Moreover, a more extensive test could be useful to assess its applicability.

We focused on reformulating the regression strategy of LIME. Observations from the Shapley theory suggests that another formulation that gives uniform weight to all strata is also possible, but it was not considered in this paper, and further investigations are needed. The goal of the proposed methodology is to avoid the undersampling of Y. In addition, the work of (Haberman 1975) proves various results and bounds between β and $\widehat{\beta}$, which could be explored further. The formulas were formulated assuming no regularization factor: however, since the main changes are in the sampling strategy, it should be possible to extend these results to ridge regression. The (briefly introduced) mixed model could also be used instead of randomly selecting the strata from a uniform distribution in the proposed algorithm. As a future work, we plan to reformulate LIME equations to better follow the neighborhood locality, which is not captured by sampling from the binomial distribution, as described in the "Limitations" section and illustrated in Fig. 2.

Availability The LIME Image with stratified sampling is available at: https://github.com/rashidrao-pk/lime_stratified All code needed to replicate the experiments (including the *requirements.txt* with the library versions used) can be found at: https://github.com/rashidrao-pk/lime-stratified-examples

Acknowledgments

This work has received funding from the European Union's Horizon 2020 research and innovation program ECSEL Joint Undertaking (JU) under Grant Agreement No. 876487, NextPerception project. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the nations involved in the mentioned projects. The work reflects only the authors' views; the European Commission is not responsible for any use that may be made of the information it contains.

References

Addison Howard, W. K., Eunbyung Park. 2018. ImageNet Object Localization Challenge. https://kaggle.com/competitions/imagenet-object-localization-challenge.

Bodria, F.; Giannotti, F.; Guidotti, R.; Naretto, F.; Pedreschi, D.; and Rinzivillo, S. 2023. Benchmarking and survey of explanation methods for black box models. *Data Mining and Knowledge Discovery*, 1–60.

DuMouchel, W. H.; and Duncan, G. J. 1983. Using sample survey weights in multiple regression analyses of stratified samples. *Journal of the American Statistical Association*, 78(383): 535–543.

Garreau, D.; and Luxburg, U. 2020. Explaining the explainer: A first theoretical analysis of LIME. In *Int. Conf. on artificial intelligence and statistics*, 1287–1296. PMLR.

Garreau, D.; and Mardaoui, D. 2021. What does LIME really see in images? In *Int. Conf. on machine learning*, 3620–3629. PMLR.

Guidotti, R.; Monreale, A.; Ruggieri, S.; Turini, F.; Giannotti, F.; and Pedreschi, D. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5): 1–42.

Haberman, S. J. 1975. How much do Gauss-Markov and least square estimates differ? A coordinate-free approach. *The Annals of Statistics*, 982–990.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conf. on computer vision and pattern recognition*, 770–778.

Konijn, H. S. 1962. Regression analysis in sample surveys. *Journal of the American Statistical Association*, 57(299): 590–606.

Liang, Y.; Li, S.; Yan, C.; Li, M.; and Jiang, C. 2021. Explaining the black-box model: A survey of local interpretation methods for deep neural networks. *Neurocomputing*, 419: 168–182.

Lundberg, S. M.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Mardaoui, D.; and Garreau, D. 2021. An analysis of LIME for text data. In *Int. Conf. on Artificial Intelligence and Statistics*, 3493–3501. PMLR.

Mitchell, R.; Cooper, J.; Frank, E.; and Holmes, G. 2022. Sampling permutations for Shapley value estimation. *The Journal of Machine Learning Research*, 23(1): 2082–2127.

Monderer, D.; and Samet, D. 2002. Variations on the Shapley value. *Handbook of game theory with economic applications*, 3: 2055–2076.

Ng, C. H.; Abuwala, H. S.; and Lim, C. H. 2022. Towards more stable LIME for explainable AI. In *2022 Int. Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, 1–4. IEEE.

Okhrati, R.; and Lipani, A. 2021. A multilinear sampling algorithm to estimate Shapley values. In *25th Int. Conf. on Pattern Recognition (ICPR)*, 7992–7999. IEEE.

Owen, G. 1972. Multilinear extensions of games. *Management Science*, 18(5-part-2): 64–79.

Rao, T. J. 1977. Optimum Allocation of Sample Size and Prior Distributions: A Review. *Int. Statistical Review*, 45(2): 173–179.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. Why should I trust you? Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD int. Conf.*, 1135–1144.

Rozemberczki, B.; Watson, L.; Bayer, P.; Yang, H.-T.; Kiss, O.; Nilsson, S.; and Sarkar, R. 2022. The Shapley value in machine learning. *IJCAI, arXiv:2202.05594*.

Sevillano-García, I.; Luengo, J.; Herrera, F.; et al. 2022. REVEL Framework to Measure Local Linear Explanations for Black-Box Models: Deep Learning Image Classification Case Study. *Int. Journal of Intelligent Systems*, 2023.

Shankaranarayana, S. M.; and Runje, D. 2019. ALIME: Autoencoder based approach for local interpretability. In *Intelligent Data Engineering and Automated Learning (IDEAL)*, 454–463. Springer.

Shi, S.; Du, Y.; and Fan, W. 2020. An extension of LIME with improvement of interpretability and fidelity. *arXiv preprint arXiv:2004.12277*.

Vedaldi, A.; and Soatto, S. 2008. Quick shift and kernel methods for mode seeking. In *Computer Vision–ECCV 2008, Proceedings, Part IV 10*, 705–718. Springer.

Vermeire, T.; Brughmans, D.; Goethals, S.; de Oliveira, R. M. B.; and Martens, D. 2022. Explainable image classification with evidence counterfactual. *Pattern Analysis and Applications*, 25(2): 315–335.

Visani, G.; Bagli, E.; Chesani, F.; Poluzzi, A.; and Capuzzo, D. 2022. Statistical stability indices for LIME: Obtaining reliable explanations for machine learning models. *Journal of the Operational Research Society*, 73(1): 91–101.

Zhang, Y.; Song, K.; Sun, Y.; Tan, S.; and Udell, M. 2019. "Why Should You Trust My Explanation?" Understanding Uncertainty in LIME Explanations. *ICML*.