bLIMEy: Surrogate Prediction Explanations Beyond LIME

Kacper Sokol
Department of Computer Science
University of Bristol
Bristol, United Kingdom
K.Sokol@bristol.ac.uk

Alexander Hepburn
Department of Engineering Mathematics
University of Bristol
Bristol, United Kingdom
ah13558@bristol.ac.uk

Raul Santos-Rodriguez
Department of Engineering Mathematics
University of Bristol
Bristol, United Kingdom
enrsr@bristol.ac.uk

Peter Flach
Department of Computer Science
University of Bristol
Bristol, United Kingdom
Peter.Flach@bristol.ac.uk

Abstract

Surrogate explainers of black-box machine learning predictions are of paramount importance in the field of eXplainable Artificial Intelligence since they can be applied to any type of data (images, text and tabular), are model-agnostic and are post-hoc (i.e., can be retrofitted). The Local Interpretable Model-agnostic Explanations (LIME) algorithm is often mistakenly unified with a more general framework of surrogate explainers, which may lead to a belief that it is the solution to surrogate explainability. In this paper we empower the community to "build LIME yourself" (bLIMEy) by proposing a principled algorithmic framework for building custom local surrogate explainers of black-box model predictions, including LIME itself. To this end, we demonstrate how to decompose the surrogate explainers family into algorithmically independent and interoperable modules and discuss the influence of these component choices on the functional capabilities of the resulting explainer, using the example of LIME.

1 Introduction

Local Interpretable Model-agnostic Explanations (LIME) [10] is a popular technique for explaining predictions of black-box machine learning models. It greatly improves on *surrogate explanations* [1] by introducing *interpretable data representations*, hence making it applicable to image and text data in addition to tabular data. Images can be represented as a collection of super-pixels and translated into a binary on/off vector indicating which super-pixel stays the same and which is occluded (removed). Equivalently, text can be represented as a bag of words translated into a similar on/off binary vector. Furthermore, LIME can be used as a fast approximation of SHapley Additive exPlanations (SHAP) [8] as the latter is computationally expensive – the cost of providing various guarantees for the produced explanations. However, the adoption of LIME is limited since it is provided as a monolithic explainability tool with little room for customisation when, in reality, it is just one possible realisation of the highly modular *surrogate explanations* framework. We argue that allowing the user to make informed choices and build a custom surrogate explainer that is designed for a specific task can greatly improve the quality of produced explanations, therefore warrant a wider adoption of surrogate explanations.

2019 Workshop on Human-Centric Machine Learning (HCML 2019).
33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Since surrogate explanations are model-agnostic, they can be applied to any predictive system, hence have a high impact in the field of eXplainable Artificial Intelligence if they become accessible, accountable and accurate. According to the "no free lunch" theorem, a single solution can never perform better than all the other approaches across the board. However, by allowing the user to take advantage of surrogate explanations' modularity, therefore customising them for a problem at hand, we can achieve the best possible explanation that the surrogate explainers family can offer. We can further improve the quality of the explanations by educating the users on the possible component choices, their properties, influence on the explanations, advantages and caveats.

To the best of our knowledge, LIME is the only available surrogate explainability tool and modifying its default behaviour often requires tinkering with LIME's source code what may discourage some of the practitioners. We address these shortcomings by taking advantage of surrogate explanations' modularity to create a unified algorithmic framework for building this type of explainers, which we call **bLIMEy** – **b**uild **LIME y**ourself. A range of possible algorithms can be used for each module – with the choices discussed in Section 3 – creating a suite of customisable surrogate explainers. Their varying capabilities and restrictions greatly influence the resulting surrogate explainer, therefore each of them should be accompanied by a critical discussion and usage suggestions. To this end, we have decomposed the surrogate explanation framework into independent algorithmic components. We implemented a choice of algorithms for each of them in Python under the BSD 3-Clause open source licence, therefore allowing for their commercial use. Our implementation is accompanied by a "how-to" guide[1] outlining how to compose custom surrogate explainers and discussing pros and cons of selected component choices. It is also capable of recreating the LIME algorithm for tabular, image and text data in a way that mitigates most of its issues reported in the literature [2, 5, 7].

Our research on surrogate explanations [1] was inspired by manuscripts investigating the instability and sources of randomness [2, 5] in LIME explanations[2] [10], which could not pinpoint the root cause of this undesired and detrimental behaviour. Laugel et al. [7] attempted to "fix" LIME for tabular data by replacing its sampling method with an explicitly local sampler, however their experiments used LIME with disabled discretisation (responsible for generating interpretable data representation), therefore unintentionally compromising the integrity of the algorithm making the two methods incomparable and the improvements not applicable to more general cases beyond the specific ones presented in their research. Henin and Le Métayer [4] introduced a unified (theoretical) framework that allows for systematic comparison of black-box explainers by characterising them alongside two dimensions: *data sampling* and *explanation generation*. In contrast, our (practical) approach is focused on algorithmic and implementation aspects of the *surrogate* subset of the black-box explainers family and extends Henin and Le Métayer's decomposition with a third dimension – *interpretable representation* – therefore bridging the gap between LIME and surrogate explanations [1].

2 bLIMEy: Modular Surrogate Explanations

Before delving into the bLIMEy framework we encourage the reader to consult Appendix A1 for an overview of the LIME explainer architecture. The bLIMEy framework decomposes surrogate explanations of a black-box model prediction for a selected data point into three distinct steps:

Interpretable Data Representation Transformation (possibly bidirectional) from the original data domain (i.e., feature space) into an interpretable domain (and back). This step is optional for tabular data but required for image and text data. Interpretable domains tend to be *binary vectors* encoding presence or absence of human-comprehensible characteristic in the data.

Data Sampling Data augmentation (sampling) in the neighbourhood of the data point selected to be explained. For images and text data sampling must be performed in the interpretable domain while for tabular data it can be done in either of the domains. Next, the sampled data must be predicted with the black-box model. If the data were sampled in the interpretable domain, they need to be *reverted* back to the original representation to complete this task.

Explanation Generation An inherently interpretable model is trained on the locally sampled data, which is used to explain the selected data point. If an interpretable (binary) representation is used, XNOR[3] can be applied between the selected data point and the sampled data

[1] `https://fat-forensics.org/how_to/transparency/tabular-surrogates.html`

[2] Produced with LIME's official open source implementation: `https://github.com/marcotcr/lime`.

[3] 1 if the same and 0 if different.

to focus the local model on presence and absence of these interpretable characteristics. The interpretable features for which the value of the selected data point is 0 can be safely removed to reduce the dimensionality. This task enforces the locality of the sampled data and introduces sparsity to the explanation[4]. Sparsity can be enforced even further by applying a dimensionality reduction technique. To further enforce the locality of the explanation, when training the local model the sampled data can be weighted based on a kernelised distance between the chosen data point and the sampled data (in either representation).

3 Discussion

When building a surrogate explainer every module choice may limit its overall functionality and the range of algorithms supported by other modules. Here, we discuss (often unintended) consequences of choosing a particular algorithm for each module and exemplify how such a choice has affected the explainer using the example of LIME. We support our discussion with an empirical comparison of various data samplers (Appendix A2) and an illustration of how a tree-based surrogate improves over a linear one, i.e., LIME, for tabular data without an interpretable representation (Appendix A3).

To avoid unnecessary randomness affecting the explanations produced by surrogate explainers, the data transformation from their original domain $\mathcal{X}$ into an interpretable representation $\mathcal{X}'$ must be **bijective** – the mapping from $\mathcal{X}$ to $\mathcal{X}'$ has to be a *one-to-one correspondence* – and it must have a corresponding and uniquely defined **inverse function** – a data point in $\mathcal{X}'$ can be translated into a unique data point in $\mathcal{X}$. In LIME, the interpretable representation for both image and text data satisfies these two requirements. A sentence can be easily represented as a binary vector indicating presence or absence of unique words in that sentence and such a binary vector can be transformed into (bag-of-words representation of) a sentence. Similar reasoning applies to images where a binary vector indicates whether a super-pixel (large, non-overlapping chunks of an image) in an image should have the same pixel values as the original image or be occluded with an arbitrary patch or a solid colour. For tabular data, an interpretable representation is achieved by discretisation and binarisation (one-hot encoding), in which case bijection is preserved but the inverse function is ill-defined. While the binarisation of categorical features (there is no need for discretisation) is invertible, numerical features that have been discretised by binning (and binarised) cannot be uniquely reconstructed into their original representation $\mathcal{X}$. LIME resolves this by sampling from a normal distribution (with clipping at bin boundaries) fitted to each numerical bin for each sampled data point to reconstruct it in the original domain ($\mathcal{X}$) – the unidentified source of randomness reported by Fen et al. [2].

This undesired behaviour for tabular data is a consequence of transforming the data into their interpretable representation $\mathcal{X}'$ first and then sampling. While such order is required for image and text data – it would be meaningless to sample from a grid of pixels or a sequence of characters respectively – it is not compulsory for tabular data, therefore providing an opportunity to avoid the "reverse sampling" step. By sampling first (in $\mathcal{X}$) and transforming the tabular data into an interpretable representation later ($\mathcal{X}'$), the inversion of the latter step is no longer required since both of the representations are available. This order of operations, however, requires the sampling procedure to be "as much local as possible" since sampling from the interpretable domain implicitly introduces the locality[5]. While any sort of sampling should suffice in the interpretable domain (as long as the XNOR filtering is performed – see the next paragraph) a local sampling method, e.g., MixUp [12] or Growing Spheres [6], is required when it is performed in the original data domain (for tabular data) – see the results shown in Appendix A2. Furthermore, sampling in this domain requires the sampling method to produce data points that are assigned more than one class (or significantly different class probabilities for probabilistic models) by the black-box model, otherwise a *meaningful* local surrogate model cannot be fitted. Given the random nature of the sampling procedure, the only way to ensure reproducible explanations is to always have the same local sample, which can only be achieved by fixing the random seed.

While reducing the dimensionality of the interpretable domain for image and text data is detrimental for the explanation – e.g., "black holes" in images and missing words in sentences – it is recommended

[4]The XNOR operation and dropping 0-valued interpretable features do not affect image and text data as the selected data point is always represented as an all-1 binary vector in the interpretable domain.

[5]For text data this is sampling from within the same sentence by leaving the words intact or removing them and for images this is modifying the original image by occluding its parts. For tabular data LIME achieves the sample locality in this step by applying the XNOR operation explained in the next paragraph.

for tabular data to reduce the number of "important factors" presented to the explainee. When an interpretable representation is not used for tabular data, dimensionality reduction should be considered a necessity. In LIME, sparsity (and locality) of an explanation for tabular data is partially achieved – when using an interpretable representation – by dropping all of the interpretable features for which the feature value of the explained data point in the interpretable domain is 0. This operation is equivalent to keeping only the categorical feature values and numerical feature bins in which the explained data point resides (the aforementioned implicit locality for tabular data). For example, for two numerical features (x_1, x_2) and their interpretable representations $(x_1 < 2, 2 \leq x_1 < 7, 7 \leq x_1)$ and $(x_2 < 0, 0 \leq x_2)$ and an explained data point $(4, 2)$ – $(0, 1, 0, 0, 1)$ in the interpretable representation – only $2 \leq x_1 < 7$ and $0 \leq x_2$ dimensions would be preserved. This step combined with transforming the interpretable feature space by applying the XNOR operation between the explained data point and the sampled data is **required** for proper functioning of LIME since its goal is to explain a prediction by quantifying the (positive or negative) effect of changing any of the interpretable feature ranges within which the explained data point resides, hence the choice of a *linear model* for the local surrogate and the use of its coefficients as the explanation. Therefore, the question that the LIME explanation tries to answer is whether for the current black-box classification of the selected data point the value of x_1 between 2 and 7 has a positive or a negative effect when compared to x_1 being outside of this range. Similarly, for a particular classification of an image or a sentence, does a given super-pixel or a word have a positive or a negative effect, i.e., would removing this super-pixel or word change the black-box classification outcome. Thereby using LIME for tabular data without an interpretable representation forfeits the locality of the sample introduced by applying the XNOR feature transformation in which case the explainer relies purely on kernelised distance weighting and normal data sampling around the explained data point (albeit with sampling variance for each feature calculated based on the whole data set, which decreases the locality effect) to induce the explanation locality. An interpretable representation for tabular data may be skipped altogether in which case other modules of the surrogate explainer, like the data sampler (cf. Appendix A2), should guarantee the locality of an explanation.

The surrogate model can be trained in a number of different ways: as a regressor of the probabilities outputted by the black-box model (as in LIME), in which case it has to model (explain) a single class selected by the user; or as a classifier when the black-box model is a thresholded probabilistic model or a classifier. Another choice is the training scheme: the surrogate model can either be trained as a multi-class or one-vs-rest predictor, with the latter approach being required for surrogate regressors of black-box probabilistic predictors. The choice of a surrogate model is also important; if local feature importance (or interpretable feature influence) is desired, a linear model is a good pick as long as all of the features are normalised to the same range (LIME satisfies this by using the interpretable binary representation or otherwise explicitly normalising the features) and these features are "reasonably" independent. A lack of normalisation causes the feature weights to be *incomparable*, therefore rendering the explanation uninformative. While explainability of linear models is limited to feature importance, a different type of an explanation – logical conditions outlining the behaviour of a black-box model in the neighbourhood of the selected data point – can be generated with a surrogate decision tree (cf. Appendix A3). The selection here should be motivated by the desired type of the explanation – e.g., "Why class A?" vs "Why class A and not B?" – and its format – a feature importance bar plot vs a conjunction of logical conditions – which are problem-dependant.

4 Conclusions and Future Work

In this paper we introduced bLIMEy: a modular algorithmic framework for building custom surrogate explainers of black-box predictions. We discussed dangers associated with algorithmic choices for each of its modules and showed how to avoid common pitfalls. bLIMEy is accompanied by an open source implementation that includes a selection of algorithms for every module of the framework, therefore empowering the community to build surrogate explainers customised to the task at hand.

In the future we will investigate the behaviour of various surrogate models in high-dimensional spaces and design a range of metrics to measure the quality and stability of surrogate explanations. We will provide one measure for each of the following three competing objectives: (1) local approximation of the closest decision boundary; (2) the ability to mimic the black-box model locally; and (3) the global faithfulness of the local surrogate model, which will engender trust in the explanations and mitigate the need for user studies, which lack universally agreed objective and often entail confirmation bias.

References

[1] Mark Craven and Jude W Shavlik. Extracting tree-structured representations of trained networks. In *Advances in neural information processing systems*, pages 24–30, 1996.

[2] Hui Fen, Kuangyan Song, Madeilene Udell, Yiming Sun, Yujia Zhang, et al. Why should you trust my interpretation? understanding uncertainty in lime predictions. *arXiv preprint arXiv:1904.12991*, 2019.

[3] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2): 179–188, 1936.

[4] Clement Henin and Daniel Le Métayer. Towards a generic framework for black-box explanations of algorithmic decision systems. *IJCAI 2019 Workshop on Explainable Artificial Intelligence (XAI)*, 2019.

[5] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Faithful and customizable explanations of black box models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. ACM, 2019.

[6] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Comparison-based inverse classification for interpretability in machine learning. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations*, pages 100–111. Springer International Publishing, 2018. ISBN 978-3-319-91473-2.

[7] Thibault Laugel, Xavier Renard, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Defining locality for surrogates in post-hoc interpretablity. *2018 ICML Workshop on Human Interpretability in Machine Learning (WHI 2018)*, 2018.

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

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] 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, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.

[11] Kacper Sokol, Raul Santos-Rodriguez, and Peter Flach. FAT Forensics: A Python Toolbox for Algorithmic Fairness, Accountability and Transparency. *arXiv preprint arXiv:1909.05167*, 2019.

[12] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=r1Ddp1-Rb`.

Appendix

All of the results presented in this appendix can be reproduced with a Jupyter Notebook hosted in the bLIMEy's GitHub repository[6]. This notebook can be executed online using Binder[7] by following the URL placed in its top cell. The experiments were done using FAT Forensics [11] – an open source package implementing various fairness, accountability and transparency algorithms. The description of each function used for these experiments can be found in the API[8] documentation of FAT Forensics[9].

A1: LIME Overview

Before discussing the LIME algorithm we examine the concept of LIME's interpretable data representations. For text data the interpretable representation is achieved by encoding a sentence as a bag of words and representing it as a binary vector, which indicates presence and absence of unique words. Such interpretable word vectors can then be represented as sentences by removing words for which the value in this binary vector has been changed to 0. The interpretable representation for images is generated by dividing images into super-pixels – non-overlapping segments – each one encompassing a part of the image that represents a concept (e.g., an object) meaningful to humans. Such interpretable image vectors can be then represented as images by occluding the super-pixels for which the value in this binary vector has been set to 0. Finally, tabular data can be transformed into the interpretable representation by one-hot encoding the categorical features and binning the numerical ones. Such interpretable tabular data vectors can be then represented in the original feature domain by altering the feature values accordingly to the changes made in the binary vector. This important concept of interpretable data representations enables LIME to explain data (i.e., raw features such as pixel values for images) or their representations (internally used by black-box predictive models, such as high-dimensional word embeddings) that are inherently human-incomprehensible.

Therefore, for a data point $x \in \mathcal{X}$ to be explained and an arbitrary black-box probabilistic predictive model $f : \mathcal{X} \to \mathcal{Y}$ the default LIME algorithm proceeds as follows:

1. Find the human interpretable representation $x' \in \mathcal{X}'$ of the data point x chosen to be explained, where $\mathcal{X}'$ denotes the interpretable domain.
2. Sample data from the interpretable domain $\mathcal{X}'$. For image and text data this is done by uniformly replacing 1's in x' with values from $\{0, 1\}$ set to get new data points in the "neighbourhood" of x, e.g., by randomly occluding super-pixels in an image or removing words from a sentence. Tabular data is first discretised into a representation where categorical features are left unchanged and each numerical feature is transformed into a categorical feature that indicates the numerical bin (interpretable representation) to which this feature belongs, e.g, $\hat{x}_3 = 1$ for $\{(-\infty, 0.5), [0.5, 1.3), [1.3, \infty)\}$ bins indicates $x_3 \in [0.5, 1.3)$. This representation ($\widehat{\mathcal{X}}$) is used for tabular data sampling to avoid assigning a sample to two different bins for a single feature, what could have happened had the sampling been performed in the binary representation ($\mathcal{X}'$) – where each bin for each feature is represented as a separate binary feature, e.g., $(x'_{3-0}, x'_{3-1}, x'_{3-2})$ sampled in a binary domain and resulting in a $(0, 1, 1)$ vector would give $x_3 \in [0.5, 1.3)$ and $x_3 \in [1.3, \infty)$. After the sampling step the tabular data is transformed into the binary interpretable representation $\mathcal{X}'$.
3. Invert the representation of the sampled data from $\mathcal{X}'$ to $\mathcal{X}$ and predict their probability for a selected class c with the black-box model f. Usually, c is selected to be the class assigned to the explained data point x by the black-box model f.
4. Drop all of the interpretable features for which the value of the explained data point (in the interpretable domain) is 0, therefore creating a new representation $\widetilde{\mathcal{X}}$ – this introduces sparsity of the explanation and enforces its locality (see Section 3 for more details).
5. Calculate the distances between the sampled data and the explained data point in $\widetilde{\mathcal{X}}$ and kernelise them (using the exponential kernel) to serve as proximity/similarity scores used to

[6] `https://github.com/So-Cool/bLIMEy/tree/master/HCML_2019`

[7] `https://mybinder.org`

[8] Application Programming Interface

[9] `https://fat-forensics.org/api.html`

weight the sampled data during the training of the local surrogate model (to enforce locality of the explanation).

6. Compute logical XNOR between the new interpretable representation $\widetilde{\mathcal{X}}$ of the explained instance $\tilde{x}$ and all of the sampled data points $\tilde{x}_{\text{sampled}}$ to create a data set $\widetilde{\mathcal{X}}_{\text{XNOR}}$ which prescribes the effect of change of a data point in the interpretable domain on its classification outcome (see Section 3 for more details).
7. Use K-LASSO to further limit the number of features used in the explanation and train a linear regression on $\widetilde{\mathcal{X}}_{\text{XNOR}}$ using the black-box predictions computed before as the target (i.e., probabilities of the previously selected class c). The coefficients of this model (feature weights) are used to interpret the (positive or negative) importance of each human-comprehensible feature.

For more details please consult the LIME manuscript [10] and its official open source implementation[10].

A2: Comparison of Data Samplers for Tabular Data

To show the behaviour of different data sampling methods we use the Iris data set [3]. We plot the data alongside two dimensions – *sepal length (cm)* on the x-axis and *sepal width (cm)* on the y-axis – to facilitate easy visual comparison. The three colours visible in the plots represent the three classes of the Iris data set: setosa, virginica and versicolor. The data set plotted alongside these two features with the markers colour-coded based on the ground truth annotation is shown in Figure 1. The background shading in this plot indicates the decision boundary of the underlying black-box model (a Random Forest Classifier trained with scikit-learn [9]).

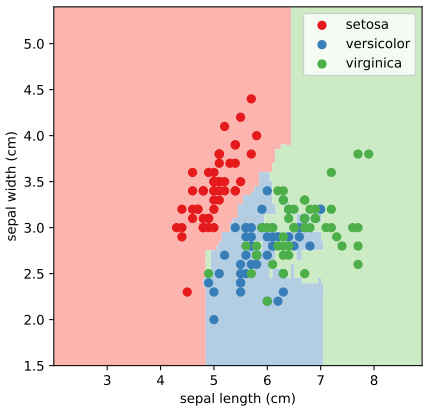
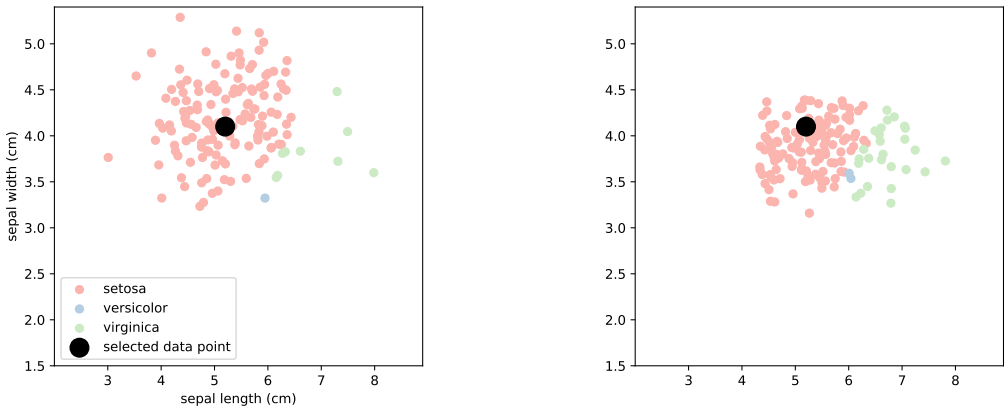


Figure 1: The Iris data set plotted alongside *sepal length (cm)* on the x-axis and *sepal width (cm)* on the y-axis with the markers colour-coded based on the ground truth annotation. The background shading represents the decision boundary of the underlying black-box model (a random forest classifier).

We initialise each of the samplers with the full Iris data set and generate 150 data points around the selected instance: the black dot. Figure 2 shows the importance of choosing an appropriate sampling method when building a surrogate explainer. Some of the data samplers may have difficulties locating the closes decision boundary (see, for example, Figure 2a and 2b) when the selected data point is far from any decision boundary of the black-box model (which may be common in high-dimensional spaces due to the curse of dimensionality), therefore generating data for which fitting a local surrogate model may not be possible. Another important aspect of the sampled data is their clear class imbalance, which needs to be accounted for during the training procedure of the surrogate model.

[10] `https://github.com/marcotcr/lime`



(a) Normal.

(b) Truncated normal.

(c) MixUp.

(d) Normal Class Discovery.

Figure 2: The effect of different sampling algorithms on the locality of the sample for the Iris data set plotted along *sepal length (cm)* on the x-axis and *sepal width (cm)* on the y-axis. The black dot is the explained data point for which the sample is generated. Red, blue and green dots are the predictions (the three classes of the Iris data set) assigned by the underlying black-box model (cf. Figure 1).

A3: Decision Tree-based Surrogate Explainer for Tabular Data

To show the importance of selecting a good surrogate model and the difference in explanations that it can produce we explain a carefully selected data point from the two moons data set. The two moons data set – shown in Figure 3 and generated with scikit-learn[11] – is a synthetic 2-dimensional, binary classification data set with a complex decision boundary. It is suitable for this type of experiments as depending on which data point is chosen the resulting explanations can be quite diverse.

Figures 4 and 5 show two variants of surrogate explainers – based on a linear and a decision tree model respectively – built for the data point marked with a black dot. It is clear that for complex decision boundaries a tree-based approach is superior. In addition to better approximating the local decision boundary (of the underlying black-box random forest classifier), a tree-based local surrogate generates locally-faithful interpretable representation from the feature splits learnt by the tree, which can be used to convey the explanation as a conjunction of logical conditions in high-dimensional spaces where visualisations are impossible.

[11] `https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_moons.html`

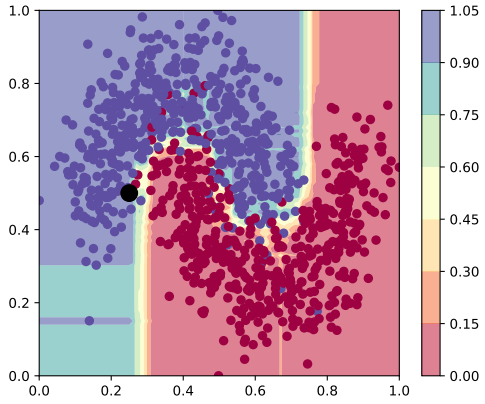


Figure 3: The two moons data set with 1000 samples. The colour of the marker indicates the ground truth label of each data point and the background shading depicts the decision boundary of the underlying black-box model (a random forest classifier). Since the model is probabilistic the colour-bar (placed to the right of the plot) provides a legend for the predicted probabilities of the blue class. The black dot represents the data point that will be explain with local surrogates (see Figures 4 and 5).

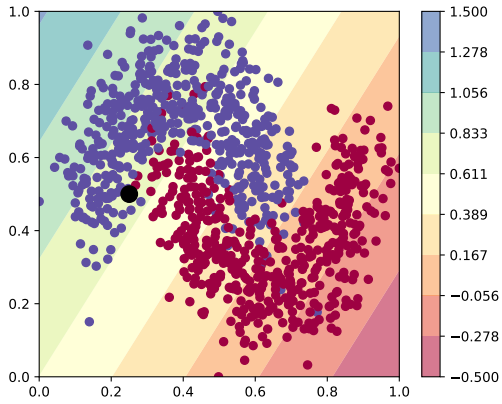


Figure 4: Linear surrogate explainer for the selected data point (black dot) – equivalent to LIME without discretisation. The background shading represents the predicted value from the local ridge regression model (the value encoding is given in the colour-bar). The local regression model is trained to predict the probability of belonging to the blue class (outputted by the black-box model), hence the predicted values may be outside of the expected $[0, 1]$ range. If the surrogate's threshold is set at 0.5, the yellow bar would be partially predicted as blue, therefore incorrectly classifying the upper left part of the red cloud of points. The importance of the x-axis feature is -1.10 and the importance of the y-axis feature is 0.69. Since an interpretable representation was not used, these values do not carry any particular meaning.

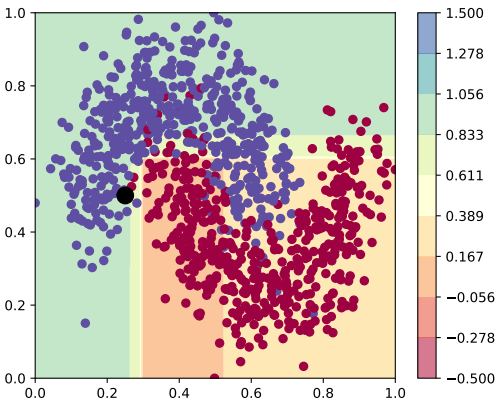


Figure 5: Decision tree-based surrogate explainer for the selected data point (black dot). The background shading represents the predicted value from the local decision tree regression model (the value encoding is given in the colour-bar). The local decision tree model is trained to predict the probability of belonging to the blue class (outputted by the black-box model). The green and light green areas have high probability of the blue class, therefore giving a good approximation of the local decision boundary, which is fairly complex for the selected data point. The orange and yellow blocks have low probability of the blue class, therefore providing a precise approximation of the red class. A possible explanation that can be derived from the local decision tree is: it is the blue class for the x-axis feature ≤ 0.265 or the y-axis feature > 0.609; and it is the red class for the y-axis feature ≤ 0.609 and the x-axis feature bounded between $(0.295, 0.528]$ – such rules can be produced for problems beyond 2-dimensions, which cannot be easily visualised.