Intepretable Local Explanations Through Genetic Programming

Hayden Andersen
andershayd@ecs.vuw.ac.nz
Victoria University of Wellington
Wellington, New Zealand

Andrew Lensen
andrew.lensen@ecs.vuw.ac.nz
Victoria University of Wellington
Wellington, New Zealand

Will N. Browne
will.browne@qut.edu.au
Queensland University of Technology
Brisbane, Australia

Yi Mei
yi.mei@ecs.vuw.ac.nz
Victoria University of Wellington
Wellington, New Zealand

ABSTRACT

As machine learning models become increasingly prevalent in everyday life, there is a growing demand for explanation of the predictions generated by these models. However, most models used by companies are black-boxes in nature, without the capacity to provide explanations to users. This reduces public trust in these models, and exists as a barrier to adoption of machine learning. Research into providing explanations to users has shown that local explanation techniques provide more acceptable explanations to users than attempting to explain an entire model, as a user often does not need to understand the entirety of a model.

This work builds on prior work in the field to produce a competitive method for high-fidelity local explanations utilising genetic programming. Two different data representations targeted towards both users with and without machine learning experience are evaluated.

The experimental results show comparable fidelity to the state-of-the art, while exhibiting more comprehensible explanations due to including fewer features in each explanation. The method enables decomposable explanations that are easy to interpret, while still capturing non-linear relationships in the original model.

KEYWORDS

Explainable AI, Machine learning, Genetic programming

ACM Reference Format:
Hayden Andersen, Andrew Lensen, Will N. Browne, and Yi Mei. 2024. Intepretable Local Explanations Through Genetic Programming. In *Genetic and Evolutionary Computation Conference (GECCO '24 Companion), July 14–18, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3638530.3654370

1 INTRODUCTION

Explainable AI (XAI) is a field of AI research that seeks to understand and explain how AI and machine learning (ML) models are making decisions. There are a number of reasons why these explanations are important, from users and lawmakers trusting these decisions to the designers of the models ensuring the models are relying on real causal relationships and not incidental connections.

The scope that an XAI method can focus on to provide explanations can take two major forms. The first is global explanation, which aims to explain an entire model to the user. The second is local explanation, which instead only explains single predictions or groups of predictions. While both of these scopes provide different benefits, local explanations are more suited to explaining machine learning models to end-users as a user does not need to understand the entire model: they only care what is relevant to their specific situation [6].

One avenue to produce a local explanation is through a local surrogate model: training a new, more interpretable model that is used as the explanation for the original prediction. The current standard of these is LIME [8], which explains a prediction by training a locally weighted linear classifier as the surrogate model. One major limitation of LIME is that the use of a linear model limits the similarity to the original model that can be reached. In addition to this, LIME utilises a feature selection method that will always select a set number of features from the data. Hence, it can often select either too many features, producing a needlessly complex explanation, or too few features, producing an explanation that does not match enough to the original prediction.

There are a number of challenges in providing optimal local explanation models for predictions. First, the model must be able to capture enough of the behaviour of the original model to give an accurate picture of the prediction. Second, the model must be able to be understood by a non-expert user or be able to be explained in such a way that it provides an acceptable explanation to a non-expert user. Finally, the model should utilise as few data features as possible in order to produce an explanation that is able to be properly comprehended by a user. We posit that Genetic Programming (GP) as an explanation technique is able to meet all three of these challenges, as it is known to be an algorithm that performs well on both accuracy and interpretability [5] that will automatically reduce the considered feature set without needing explicit instruction to do so [10]. However, prior GP-based research to produce local explanations [3] has neglected to address the first challenge as a lack of localised weightings in the data generation results in a global explanation as opposed to a true local explanation.

This paper introduces a technique utilising GP to provide high fidelity local explanations for predictions made by black-box machine learning models. These explanations are locally weighted against the prediction being explained, and the tree structure allows for

GECCO '24 Companion, July 14–18, 2024, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-0495-6/24/07.
https://doi.org/10.1145/3638530.3654370

strong decompositional explanation. The trees are quantitatively shown to utilise fewer features from the data than the current standard algorithms, while exhibiting competitive fidelity. The key contributions of this work are to:

- Propose a new approach using GP to produce local explanations for predictions made by machine learning models. Unlike previous research applying GP to produce local explanations this approach is shown to work on multi-class classification problems, as most real-world applications are not binary classification.
- Demonstrate the capability of the proposed approach to produce high fidelity explanations that utilise minimal data features through quantitative comparison with LIME, the modern standard of local explanation models. Unlike LIME, this approach allows for explanation involving non-linear decision boundaries due to the flexible model structure of GP.

2 BACKGROUND

2.1 Explainable AI Evaluation

It is often a relatively simple task to determine an optimal prediction in supervised ML domains. For example, many classification algorithms will use accuracy as a comparison metric, with a higher accuracy corresponding to a better model. However, as defined by Robnik-Šikonja and Bohanec [9], in an XAI context accuracy is the ability of the explanation model to generalise to unseen data, while fidelity is defined as the ability of the explanation to match the actual predictions of the model being explained. Given these definitions, this work focuses on the fidelity of the explanations. In addition, this work focuses on comprehensibility: the ease with which a non-expert user can understand a provided explanation.

2.2 Local Surrogate Explanation Models

Given the task of providing an explanation for a model in an ML context, a common approach is to utilise a simpler, more intrinsically comprehensible model and treat this model as a surrogate to provide an explanation for the original model [7]. One reason for this is that it is a simple way to explain a completely black-box model, only needing to know inputs and outputs to produce an explanation. A local surrogate model is a surrogate model that aims to only explain a single prediction or group of predictions, rather than explaining the entire model. This allows the explanations to be considerably less complex, as they only need to explain the decision boundaries that impact that prediction [7].

Local Interpretable Model-agnostic Explanations (LIME), proposed by Ribeiro et al. [8] is one such explanation method. It is the most well-known of these methods, often representing the entire scope of local surrogate models in published works. LIME proposes a framework to produce a sparse linear model as the interpretable model around a single prediction, using a simplified representation of the original data in order to produce a more interpretable model.

There are three major steps LIME takes to produce an explanation. First, it creates a synthetic dataset of instances around the instance being predicted, and weights them based on an exponential kernel. This dataset is created as a binary dataset, with the method used to produce this dataset differing depending on the type of data being explained. Second, it selects 10 features using Lasso. Finally, it trains the linear model using these 10 features in the synthetic data, and uses this model as the surrogate explanation.

2.3 Related Work

There are two major pieces of work that aim to provide a surrogate GP model for a more complex black-box model. Evans et al. [2] proposed a method to approximate a given black-box model with a single GP tree, training the GP with the original feature values of the data and the predicted class labels from the black-box. This provides a global explanation, approximating the behaviour of the entire model using the GP tree. A drawback with this approach, however, is that the same data is used to train both the original model and the surrogate, with the only difference being that the target values are the outputs of the original model instead of the targets from the data. A more representative global explanation method would be to instead sample new data from the same distribution as the original data, as this would ensure that the GP surrogate is explaining the workings of the model and not simply retraining on the same information as the original model

Ferreira et al. [3] later proposed an algorithm aptly named Genetic Programming Explainer (GPX), producing a local explanation model by using GP to explain a single prediction. However, there are two major drawbacks to this algorithm. First, the method samples the synthetic data from around the initial prediction in a multivariate Gaussian distribution, but does not provide any weightings to these synthetic instances. This reduces the trust that the produced GP trees are locally faithful to the prediction being explained, and faces a similar problem as the global explanation method proposed by Evans et al. Secondly, GPX can only explain predictions made on binary classification tasks, as the GP implementation used will simply predict one class if the output of the GP tree is greater than 0.5 and the other if the output is less than 0.5.

3 PROPOSED METHOD

The proposed approach utilises GP to explain a single prediction made by a machine learning model. To perform this, a synthetic dataset is created, based around the instance being explained. There are two different representations used for this synthetic dataset, described below in Section 3.1. Once this synthetic dataset is created, a kernel is used to weight each synthetic instance based on their distance from the original instance being explained. There are two kernels used in this work, one based on the kernel shown in the work by Ribeiro et al. [8], and one based on the subsequent published library. The standard kernel is shown in Equation (1), where $\pi(z)$ is the weight assigned to synthetic instance z, $D(z)$ gives the Euclidean distance from z to the instance i being explained, and σ is the kernel width defined by $\sqrt{dim(i)} * 0.75$. The square root kernel is defined the same, except the result of $\pi(z)$ is then square rooted. There has been some criticism of the LIME kernel definition [4], however for a fair comparison between our proposed method and LIME the original definition is used.

$$\pi(z) = exp(-D(z)^2/\sigma^2) \tag{1}$$

The synthetic dataset and weightings are then used as the input data in a GP learning algorithm, eventually producing a final population

of trees. The fittest tree is taken from this population, providing the final explanation model. The GP algorithm used is described below in Section 3.2.

3.1 Synthetic Data Representation

There are two main approaches to data representation for the synthetic data taken in this work, one inspired by Ferreira et al. [3] and one inspired by Ribeiro et al. [8]. The first is to sample continuous data from the area around the original instance through a multivariate Gaussian distribution.

The second data representation explored is similar to the one used by Ribeiro et al. [8], representing the synthetic data as a vector of binary values. In non-technical terms, a 1 in this vector represents a feature that is similar in value to the original instance, and a 0 represents a feature that is further away in value. To create this binary data, the original continuous data is binned into distinct sets, and features are sampled randomly from each bin label. The feature is then set to 1 if it is in the same bin as the original instance, or 0 otherwise.

The final step of the synthetic feature space generation is to replace one of the generated instances with the original instance being explained, to ensure the original data is still being considered. It is worth noting that with the given kernel definition, the original instance will always be assigned the highest weight.

Once the synthetic representation of the feature space has been generated, the target labels are created. This is done through using the black-box being explained to produce predictions for these representations. For the first data representation, the raw data is used for this, and for the second data representation, the reconstruction is used. This works best with black-box models that produce a probabilistic output as it turns the prediction task for the explanation into a regression problem, which GP is better suited towards. The target labels p for the GP evolutionary process are then recorded as the predicted probability of the original class label of the instance being explained.

3.2 GP Method

A standard GP algorithm is used in this work, following an iterative procedure of selecting a given number of the fittest individuals in the population to carry into the next generation unchanged, then probabilistically choosing between mutation and crossover to create the remainder of the new population.

The fitness function used is the weighted mean squared error between the GP output and the synthetic labels, given in Equation (2). This treats the task of explaining categorical predictions as a regression problem.

$$fitness = \frac{1}{n}\sum_{i=1}^{n} \pi_i (y_i - p_i)^2 \tag{2}$$

4 EXPERIMENTAL DESIGN

4.1 Datasets

Nine datasets are used to evaluate the proposed method. Seven were sourced from the UCI machine learning repository, one (kc2) from the OpenML platform, and the final dataset (Penguins) is from an online R repository. These are all classification datasets, with numbers of features ranging from 4 to 500, numbers of instances ranging from 178 to 4400, and number of classes ranging from 2 to 7. This gives a good representation of a range of different tasks. The datasets are shown in Table 1, ordered by the approximate complexity posed by the classification task.

Table 1: Chosen datasets

Name	Features	Instances	Classes
Penguins	4	333	3
Breast Cancer Wisconsin (BCW)	10	699	2
Wine	13	178	3
Segmentation	18	2310	7
kc2	21	522	2
Steel Plates Faults (SPF)	27	1941	2
Ionosphere	33	351	2
Dermatology	34	366	6
Madelon	500	4400	2

4.2 Experiment Setup

For each combination of data representation and kernel function, the following experimental steps are taken:

(1) Train a black-box model on the provided training data. For these experiments a random forest classifier was used as an example of a performant black-box model that does not require much prior parameter tuning. However, any black-box model could be used.
(2) Select a single random instance from the data.
(3) Use the black-box model to predict the class of the instance.
(4) Produce an synthetic dataset around the chosen instance as described in Section 3.1. For the sake of this work, this dataset is always created with 1000 instances in order to provide a strong range of differently weighted values.
(5) Produce weightings for each instance using the chosen kernel function.
(6) Use the GP method described in Section 3.2 to produce an explanation model.
(7) Perform algebraic simplification on the final GP tree.
(8) Repeat from Step 2 five times, using the same black-box model but a different chosen instance.

The GP training is performed with a population size of 120, 1000 generations, 5-tournament selection, one point crossover, subtree mutation, 3-elitism, and mutation and crossover probabilities of 0.1 and 0.9 respectively. These parameters were found in initial experimentation to work well for a range of datasets. While these could be improved for each specific dataset, the goal of this work is to produce an explanation method that works without excessive tuning on the part of a user in order to find a useful explanation. As it can not be expected for an end user to need to tune these for each specific task, we have elected to keep them the same across each dataset.

To ensure the results are not biased to specific black-box models or instances chosen and to give enough data for statistical significance testing, each algorithm and dataset undergoes the above steps 30 times with 30 different random seeds.

In order to provide a fair comparison to prior work, the above steps are also repeated on the LIME method described by Ribeiro et

Table 2: GP Significant results compared to LIME

Data Representation	Kernel	Features Selected	Fidelity
Continuous	Standard	4 ↑ 3 ↓	0 ↑ 9 ↓
	Sqrt	5 ↑ 2 ↓	1 ↑ 8 ↓
Discrete	Standard	2 ↑ 2 ↓	6 ↑ 1 ↓
	Sqrt	3 ↑ 3 ↓	5 ↑ 3 ↓

al. [8]. This is different than their evaluation in the original paper, as they focused more on ensuring specific features were selected.

5 RESULTS AND DISCUSSION

5.1 Experimental Results

The results of the experiments are shown in Table 2. For each algorithm, the table shows the number of wins and losses of the GP algorithm with respect to LIME, represented respectively by a ↑ and a ↓. These wins and losses are evaluated using a Wilcoxon signed rank test, corrected across each data representation with Hommel's method to account for type-1 errors. An alpha value of 0.05 was used. The two metrics evaluated are the size of the feature set used, as a simplified functional evaluation of the comprehensibility of the explanation, and the fidelity of the explanation to the original prediction according to Equation (2).

5.2 Analysis

While not shown in the result tables for conciseness, the comparison between the two kernels warrants discussion. In terms of fidelity, the standard and square root kernels each fully outperform each other on 3 datasets in the continuous data representation, and the fidelity on the square root kernel outperforms or equals the standard kernel on all but two datasets in the discrete data representation. There is no clear pattern to which kernel selects fewer features from the data for the GP algorithm, so choice of the better kernel to use changes depending on the task.

On the continuous data LIME has a better fidelity on each dataset. However, on five datasets the GP method selects statistically fewer features than LIME to include in the explanation. These fewer selected features lead to a more comprehensible model that is easier for a user to understand. While in some cases a higher fidelity explanation would be required, it has been shown [6] that in many cases a less "correct" explanation is often preferred by a user if it is close to be true and is simpler to reason about. As in all but three cases the fidelity loss for GP is smaller than 1% of the overall range of the target value, the explanations with fewer features would in most cases be preferred by a user.

On the discrete data GP has a better fidelity than LIME on all datasets except kc2, SPF, and Madelon. In addition to this, the GP method selects fewer or a similar number features to LIME on all datasets except Ionosphere, Dermatology, and Madelon. However on both Ionosphere and Dermatology the fidelity difference is minimal, so the reduced feature set size means that LIME is the stronger method for these datasets, while on SPF the minimal fidelity difference means that GP is the stronger explanation method.

In summary, on both the continuous and discrete data representations the GP method on both kernels produces stronger explanations with fewer selected features than LIME, despite in some cases exhibiting marginally worse fidelity. This advantage reduces as the complexity of the problems increases, however, likely due to the increasing difficulty of the embedded feature selection.

6 CONCLUSIONS

In this paper we have introduced a new method that utilises GP to produce high fidelity explanations for predictions made by machine learning models. Our experiments have demonstrated that these explanations achieve similar performance to state-of-the-art methodologies for local explanations while being highly comprehensible due to utilising significantly fewer features from the data.

Future work will focus on the capability of GP to produce a diverse set of explanations, allowing different explanations to be provided to different users based on their own inherent biases and experiences. It is known that a user will be much more likely to accept an explanation if it takes these into consideration, so an optimal explanation model should do so. Our prior work [1] has shown that population-based algorithms are a strong contender to produce sets of counterfactual explanations, so this idea will be expanded into this work through niching GP algorithms.

REFERENCES

[1] Hayden Andersen, Andrew Lensen, Will Browne, and Yi Mei. 2023. Producing Diverse Rashomon Sets of Counterfactual Explanations with Niching Particle Swarm Optimization Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Lisbon, Portugal) *(GECCO '23)*. Association for Computing Machinery, New York, NY, USA, 393–401. https://doi.org/10.1145/3583131.3590444

[2] Benjamin P Evans, Bing Xue, and Mengjie Zhang. 2019. What's inside the Black-Box? A Genetic Programming Method for Interpreting Complex Machine Learning Models. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '19)*. Association for Computing Machinery, New York, NY, USA, 1012–1020. https://doi.org/10.1145/3321707.3321726

[3] Leonardo Augusto Ferreira, Frederico Gadelha Guimarães, and Rodrigo Silva. 2020. Applying Genetic Programming to Improve Interpretability in Machine Learning Models. In *2020 IEEE Congress on Evolutionary Computation (CEC)*. 1–8. https://doi.org/10.1109/CEC48606.2020.9185620

[4] Romaric Gaudel, Luis Galárraga, Julien Delaunay, Laurence Rozé, and Vaishnavi Bhargava. 2022. s-LIME: Reconciling Locality and Fidelity in Linear Explanations. In *Advances in Intelligent Data Analysis XX*, Tassadit Bouadi, Elisa Fromont, and Eyke Hüllermeier (Eds.). Springer International Publishing, Cham, 102–114.

[5] Yi Mei, Qi Chen, Andrew Lensen, Bing Xue, and Mengjie Zhang. 2023. Explainable Artificial Intelligence by Genetic Programming: A Survey. *IEEE Transactions on Evolutionary Computation* 27, 3 (2023), 621–641. https://doi.org/10.1109/TEVC.2022.3225509

[6] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267 (feb 2019), 1–38. https://doi.org/10.1016/j.artint.2018.07.007 arXiv:1706.07269

[7] Christoph Molnar. 2019. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Leanpub. https://christophm.github.io/interpretable-ml-book/

[8] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?" Explaining the Predictions of Any Classifier. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 13-17-Augu (feb 2016), 1135–1144. https://doi.org/10.1145/2939672.2939778 arXiv:1602.04938

[9] Marko Robnik-Šikonja and Marko Bohanec. 2018. *Perturbation-Based Explanations of Prediction Models*. Springer International Publishing, Cham, 159–175. https://doi.org/10.1007/978-3-319-90403-0_9

[10] Binh Tran, Bing Xue, and Mengjie Zhang. 2019. Genetic programming for multiple-feature construction on high-dimensional classification. *Pattern Recognition* 93 (2019), 404–417.