ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200387

Measurable Counterfactual Local Explanations for Any Classifier

Adam White and **Artur d'Avila Garcez**[1]

Abstract. We propose a novel method for explaining the predictions of any classifier. In our approach, local explanations are expected to explain both the outcome of a prediction and how that prediction would change if 'things had been different'. Furthermore, we argue that satisfactory explanations cannot be dissociated from a notion and measure of fidelity, as advocated in the early days of neural networks' knowledge extraction. We introduce a definition of *fidelity to the underlying classifier* for local explanation models which is based on distances to a target decision boundary. A system called CLEAR: Counterfactual Local Explanations via Regression, is introduced and evaluated. CLEAR generates *b*-counterfactual explanations that state minimum changes necessary to flip a prediction's classification. CLEAR then builds local regression models, using the *b*-counterfactuals to measure and improve the fidelity of its regressions. By contrast, the popular LIME method [17], which also uses regression to generate local explanations, neither measures its own fidelity nor generates counterfactuals. CLEAR's regressions are found to have significantly higher fidelity than LIME's, averaging over 40% higher in this paper's five case studies.

1 Introduction

Machine learning systems are increasingly being used for automated decision making. It is important that these systems' decisions can be trusted. This is particularly the case in mission critical situations such as medical diagnosis, airport security or high-value financial trading. Yet the inner workings of many machine learning systems seem unavoidably opaque. The number and complexity of their calculations are often simply beyond the capacities of humans to understand. One possible solution is to treat machine learning systems as 'black-boxes' and to then explain their input-output behaviour. Such approaches can be divided into two broad types: those providing global explanations of the entire system and those providing local explanations of single predictions. Local explanations are needed when a machine learning system's decision boundary is too complex to allow for global explanations. This paper focuses on local explanations.

Unfortunately, many explainable AI projects have been too reliant on their researchers' own intuitions as to what constitutes a satisfactory local explanation [13]. Yet the required structure of such explanations has been extensively analysed within philosophy, psychology and cognitive science. Miller[13, 12] has carried out a review of over 250 papers from these disciplines. He states that perhaps his most important finding is that explanations are counterfactual: "they are sought in response to particular counterfactual cases ... why event P happened instead of event Q. This has important social and computational consequences for explainable AI". An explanation of a classification needs to show why the machine learning system did not make some alternative (expected or desired) classification.

A novel method called **C**ounterfactual **L**ocal **E**xplanations vi**A** **R**egression (CLEAR) is proposed. This is based on the philosopher James Woodard's [22] seminal analysis of counterfactual explanation. Woodward's work derives from Pearl's manipulationlist account of causation [16]. Woodward states that a satisfactory explanation consists in showing patterns of counterfactual dependence. By this he means that it should answer a set of 'what-if-things-had-been-different?' questions, which specify how the explanandum (i.e. the event to be explained) would change if, contrary to the fact, input conditions had been different. It is in this way that a user can understand the relevance of different features, and understand the different ways in which they could change the value of the explanandum. Central to Woodward's notion is the requirement for an explanatory generalization:

> "Suppose that M is an explanandum consisting in the statement that some variable Y takes the particular value y. Then an explanans E for M will consist of (a) a generalization G relating changes in the value(s) of a variable X (where X may itself be a vector or n-tuple of variables X_i) with changes in Y, and (b) a statement (of initial or boundary conditions) that the variable X takes the particular value x."

In Woodward's analysis, X causes Y. For our purposes, Y can be taken as the machine learning system's predictions where X are the system's input features. The required generalization can be a regression equation that captures the machine learning system's local input-output behaviour. For Woodward, an explanation not only enables an understanding of why an event occurs, it also identifies changes ('manipulations') to features that would have resulted in a different outcome.

CLEAR provides counterfactual explanations by building on the strengths of two state-of-the-art explanatory methods, while at the same time addressing their weaknesses. The first is by Wachter et al. [20, 15] who argue that single predictions are explained by what we shall term as 'boundary counterfactuals' (henceforth: '*b*-counterfactuals') that state the minimum changes needed for an observation to 'flip' its classification. The second method is by Riberio et al. [17] who argue for Local Interpretable Model-Agnostic Explanations (LIME). These explanations are created by building a regression model that seeks to approximate the local input-output behaviour of the machine learning system.

In isolation, *b*-counterfactuals do not provide explanatory generalizations relating X to Y and therefore are not satisfactory explanations, as we exemplify in the next section. LIME, on the other hand,

[1] City, University of London, UK, email: Adam.White.2@city.ac.uk A.Garcez@city.ac.uk

does not measure the fidelity of its regressions and cannot produce counterfactual explanations.

The contribution of this paper is three-fold. We introduce a novel explanation method capable of:

- providing *b*-counterfactuals that are explained by regression coefficients including interaction terms;
- evaluating the fidelity of its local explanations to the underlying learning system;
- using the values of *b*-counterfactual to significantly improve the fidelity of its regressions.

When applied to this paper's five case studies, CLEAR improves on the fidelity of LIME by an average of over 40%.

Section 2 provides the background to CLEAR including an analysis of *b*-counterfactuals and LIME. Section 3 introduces CLEAR and explains how it uses *b*-counterfactuals to both measure and improve the fidelity of its regressions. Section 4 contains experimental results on five datasets showing that CLEAR's regressions have significantly higher fidelity than LIME's. Section 5 concludes the paper and discusses directions for future work.

2 Background

This paper adopts the following notation: let m be a machine learning system mapping $X \rightarrow Y$; m is said to generate prediction y for observation $\boldsymbol{x}$.

2.1 *b*-Counterfactual Explanations

Wachter et al.'s *b*-counterfactuals explain a single prediction by identifying 'close possible worlds' in which an individual would receive the prediction they desired. For example, if a banking machine learning system declined Mr Jones' loan application, a *b*-counterfactual explanation might be that 'Mr Jones would have received his loan, if his annual salary had been \$35,000 instead of the \$32,000 he currently earns'. The \$3000 increase would be just sufficient to flip Mr Jones to the desired side of the banking system's decision boundary.

Wachter et al. note that a counterfactual explanation may involve changes to multiple features. Hence, an additional *b*-counterfactual explanation for Mr Jones might be that he would also get the loan if his annual salary was \$33,000 and he had been employed for more than 5 years. Wachter et al. state that counterfactual explanations have the following form:

> "Score p was returned because variables V had values ($v_1, v_2, \ldots$) associated with them. If V instead had values ($v'_1, v'_2, \ldots$), and all other variables remained constant, score p' would have been returned"

The key problem with *b*-counterfactuals: *b*-counterfactual explanations fail to satisfy Woodward's requirement that: *a satisfactory explanation of prediction y should state a generalization relating X and Y.*

For example, suppose that a machine learning system has assigned Mr Jones a probability of 0.75 for defaulting on a loan. Although stating the changes needed to his salary and years of employment has explanatory value, this falls short of being a satisfactory explanation. A satisfactory explanation also needs to explain:

1. why Mr Jones was assigned a score of 0.75. This would include identifying the contribution that each feature made to the score.
2. how the features interact with each other. For example, perhaps the number of years employed is only relevant for individuals with salaries below \$34,000.

These requirements could be satisfied by stating an explanatory equation that included interaction terms and indicator variables. At a minimum, the equation's scope should cover a neighbourhood around $\boldsymbol{x}$ that includes the data points identified by its *b*-counterfactuals.

2.2 Local Interpretable Model-Agnostic Explanations

Ribeiro et al. [17] propose LIME, which seeks to explain why m predicts y given $\boldsymbol{x}$ by generating a simple linear regression model that approximates m's input-output behaviour with respect to a small neighbourhood of m's feature space centered on $\boldsymbol{x}$. LIME assumes that for such small neighbourhoods m's input-output behaviour is approximately linear. Ribeiro et al. recognize that there is often a trade off to be made between local fidelity and interpretability. For example, increasing the number of independent variables in a regression might increase local fidelity but decrease interpretability. LIME is becoming an increasingly popular method, and there are now LIME implementations in multiple packages including Python, R and SAS.

The LIME algorithm: Consider a model m (e.g. a random forest or MLP) whose prediction is to be explained: The LIME algorithm: (1) generates a dataset of synthetic observations; (2) labels the synthetic data by passing it to the model m which calculates probabilities for each observation belonging to each class. These probabilities are the ground truths that LIME is trying to explain; (3) weights the synthetic observations (in standardised form) using the kernel: $K(d) = \sqrt{e^{-\left(d^2/kernelwidth^2\right)}}$ where d is the Euclidean distance from $\boldsymbol{x}$ to the synthetic observation, and the default value for kernel width is a function of the number of features in the training dataset; (4) produces a locally weighted regression, using all the synthetic observations. The regression coefficients are meant to explain m's forecast y.

Key problems with LIME: *LIME does not measure the fidelity of its regression coefficients. This hides that it may often be producing false explanations.* Although LIME displays the values of its regression coefficients, it does not display the predicted values y calculated by its regression model. Let us refer to these values as *regression scores* (they are not bounded by the interval [0,1]). Ribeiro provides an online 'tutorial' on LIME, which includes an example of a random forest model of the Iris dataset[2]. As part of this paper's analysis, the LIME regression scores were captured revealing large errors. For example, in ≈20% of explanations, the regression scores differed by more than 0.4 from the probabilities calculated by the random forest model.

It might be thought that an adequate solution would be to provide a goodness-of-fit measure such as adjusted R-squared. However, as will be explained in Section 3, such measures can be highly misleading when evaluating the fidelity of the regression coefficients for estimating *b*-counterfactuals.

Another problem is that *LIME does not provide counterfactual explanations.* It might be argued that LIME's regression equations provide the basis for a user to perform their own counterfactual calculations. However, there are multiple reasons why this is incorrect. First, as will be shown in Section 3, additional functionality is necessary for generating faithful counterfactuals including the ability to

[2] https://github.com/marcotcr/lime

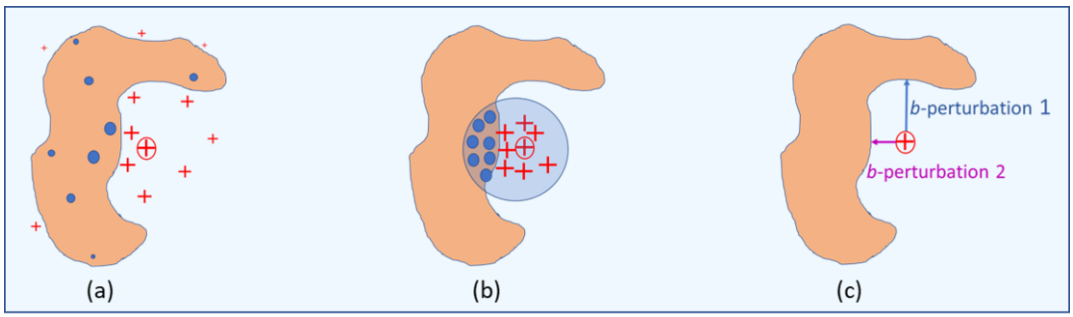

Figure 1. Toy example of a machine learning function represented by tan/blue background. The circled cross is $\boldsymbol{x}$ whose prediction is to be explained. The other crosses are synthetic observations. (a) LIME uses all synthetic observations in each regression (15,000 in this paper) with weights decreasing with distance from $\boldsymbol{x}$. (b) CLEAR selects a balanced subset of $\approx$ 200 synthetic observation. (c) shows the corresponding b-perturbations.

solve quadratic equations. Second, LIME does not ensure that the regression model correctly classifies $\boldsymbol{x}$. In cases where the regression misclassifies $\boldsymbol{x}$'s class, then any subsequent *b*-counterfactual will be false. Third, it does not have the means of measuring the fidelity of any counterfactual calculations derived from its regression equation. Fourth, LIME does not offer an adequate dataset for calculating counterfactuals. The data that LIME uses in a local regression needs to be representative of the neighbourhood that its explanation is meant to apply to. For counterfactual explanations, this extends from $\boldsymbol{x}$ to the nearest points of *m*'s decision boundary.

2.3 Other Related Work

Early work seeking to provide explanations to neural networks have been focused on the extraction of symbolic knowledge from trained networks [1], either decision trees in the case of feedforward networks [4] or graphs in the case of recurrent networks [10, 21]. More recently, attention has been shifted from global to local explanation models due to the very large-scale nature of current deep networks, and has been focused on explaining specific network architectures (such as the bottleneck in auto-encoders [8]) or domain specific networks such as those used to solve computer vision problems [3], although some recent approaches continue to advocate the use of rule-based knowledge extraction [2, 19]. The reader is referred to Guidotti et al. [6] for a recent survey.

More specifically, Lundberg et al.[11] propose SHAP, which explains a prediction by using the game-theory concept of Shapley Values. Shapley Values are the unique solution for fairly attributing the benefits of a cooperative game between players, when subject to a set of local accuracy and consistency constraints. SHAP explains a model m's prediction for observation $\boldsymbol{x}$ by first simplifying *m*'s input data to binary features. A value of 1 corresponds to a feature having the same value as in $\boldsymbol{x}$ and a value of 0 corresponds to the feature's value being 'unknown'. Within this context, Shapley Values are the changes that each simplified feature makes to *m*'s expected prediction when conditioned on that feature. Lundberg et al. derive a kernel that enables regressions where (i) the dependent variable is *m*'s (re-based) expected prediction conditioned on a specific combination of binary features (ii) the independent features are the binary features (iii) the regression coefficients are the Shapley Values. A key point for this paper is that the Shapley Values apply to the binarized features and cannot be used to estimate the effects of changing a numeric feature of $\boldsymbol{x}$ by a particular amount. They therefore do not provide a basis for estimating *b*-counterfactuals.

Ribeiro et al. [18], the authors of the LIME paper, have subsequently proposed 'Anchors: High Precision Model-Agnostic Explanations'. In motivating their new method they note that LIME does not measure its own fidelity and that 'even the local behaviour of a model may be extremely non-linear, leading to poor linear approximations'. An Anchor is a rule that is sufficient (with a high probability) to ensure that a local prediction will remain with the same classification, irrespective of the values of other variables. The extent to which the Anchor generalises is estimated by a 'coverage' statistic. For example, an Anchor for a model with the Adult dataset could be: "If Never-Married and Age $\leq$ 28 then *m* will predict '$\leq$ \$50k' with a precision of 97% and a coverage of 15%". A pure-exploration multi-armed bandit algorithm is used to efficiently identify Anchors. As with SHAP, Anchors do not provide a basis for estimating *b*-counterfactuals. Therefore neither method can be directly compared to CLEAR.

LIME has spawned several variants. For example LIME-SUP [9] and K-LIME [7] both seek to explain a machine learning system's functionality over its entire input space by partitioning the input space into a set of neighbourhoods, and then creating local models. K-LIME uses clustering and then regression, LIME-SUP just uses decision tree algorithms. LIME has also been adapted to enable novel applications, for example SLIME [14] provides explanations of sound and music content analysis. However none of these variants address the problems identified with LIME in this paper.

3 The CLEAR Method

CLEAR is based on the view that a satisfactory explanation of a single prediction needs to both explain the value of that prediction and answer 'what-if-things-had-been-different' questions. In doing this it needs to state the relative importance of the input features and show how they interact. A satisfactory explanation must also be measurable and state how well it can explain a model. It must *know when it does not know* [5].

CLEAR is based on the concept of *b*-perturbation, as follows:

Definition 5.1 Let $\min_f(\boldsymbol{x})$ denote a vector resulting from applying a minimum change to the value of one feature f in $\boldsymbol{x}$ such that $m(\min_f(\boldsymbol{x})) = y'$ and $m(\boldsymbol{x}) = y$, class(y) $\neq$ class(y'). Let $v_f(\boldsymbol{x})$ denote the value of feature f in $\boldsymbol{x}$. A ***b*-perturbation** is defined as the change in value of feature f for target class(y'), that is $v_f(\min_f(\boldsymbol{x})) - v_f(\boldsymbol{x})$.

For example, for the *b*-counterfactual that Mr Jones would have received his loan if his salary had been \$35,000, a *b*-perturbation for *salary* is \$3000. If $\boldsymbol{x}$ has p features and m solves a k-class problem

CLEAR Report: PIMA dataset

Prediction to be explained: Observation 2 has 0.72 probability of diabetes

***b*-counterfactuals**

feature	input value	actual *b*-counterfactual value
Glucose	0.24	-1.08
BMI	0.65	-0.89
Age	0.36	-0.92

Regression

prediction = 0.44 - 0.018 BloodPressure + 0.16 BMI + 0.015 Pregnancies - 0.013 SkinThickness + 0.23 Glucose - 0.12 Insulin + 0.061 DiabFunction + 0.23 Age - 0.11 Age^2 - 0.071 BMI^2 + 0.12 (Insulin*Age)

feature	estimated *b*-counterfactual value	fidelity error
Glucose	-0.72	0.37
BMI	-0.73	0.17
Age	-0.74	0.18

Figure 2. Excerpt from a CLEAR *b*-counterfactual report. In this example CLEAR uses multiple regression to explain a single prediction generated by an MLP model trained on the Pima dataset

then there are $q \leq p \times k - 1$ *b*-perturbations of $\boldsymbol{x}$; changes in a feature value may not always imply a change of classification.

CLEAR compares each *b*-perturbation with an estimate of that value, call it *estimated* *b*-perturbation, calculated using its local regression, to produce a *counterfactual fidelity error*, as follows:

$$\textit{counterfactual fidelity error} = \\ |\ \text{estimated } b\text{-perturbation} - b\text{-perturbation}\ |$$

Henceforth these will be referred to simply as 'fidelity errors'. CLEAR generates an explanation of prediction y for observation $\boldsymbol{x}$ by the following steps:

1. Determine $\boldsymbol{x}$'s *b*-perturbations. For each feature *f*, a separate one-dimensional search is performed by querying *m* starting at x, and progressively moving away from x by changing the value of f by regular amounts, whilst keeping all other features constant. The searches are constrained to a range of possible feature values.
2. Generate labelled synthetic observations (default: 50,000 observations). Data for numeric features is generated by sampling from a uniform distribution. Data for categorical features is generated by sampling in proportion to the frequencies found in the training set. The synthetic observations are labelled by being passed through *m*.
3. Create a balanced neighbourhood dataset (default is 200 observations). Synthetic observations that are near to $\boldsymbol{x}$ (Euclidean distance) are selected with the objective of achieving a dense cloud of points between $\boldsymbol{x}$ and the nearest points just beyond *m*'s decision boundaries (Figure 1). For this, the neighbourhood data is selected such that it is equally distributed across classes, i.e. approximately balanced. CLEAR was also evaluated using 'imbalanced' neighbourhood data, where only the synthetic observations nearest to $\boldsymbol{x}$ were selected. However, it was found this reduced fidelity (see Figure 3). It might be thought that a balanced neighbourhood dataset should not be used when $\boldsymbol{x}$ is far from *m*'s decision boundary; as this would lead to a 'non-local' explanation. But this misses the point that a satisfactory explanation is counterfactual, and that the required locality therefore extends from $\boldsymbol{x}$ to data points just beyond *m*'s decision boundary.
4. Perform a step-wise regression on the neighbourhood dataset, under the constraint that the regression curve should go through $\boldsymbol{x}$. The regression can include second degree terms, interaction terms and indicator variables. CLEAR provides options for both multiple and logistic regression. A user can also specify terms that must be included in a regression.
5. Estimate the *b*-perturbations by substituting $\boldsymbol{x}$'s *b*-counterfactual values from $\min_f(\boldsymbol{x})$, other than for feature f, into the regression equation and calculating the value of f. See example below.
6. Measure the fidelity of the regression coefficients. Fidelity errors are calculated by comparing the actual *b*-perturbations determined in step 1 with the estimates calculated in step 5.
7. Iterate to best explanation. Because CLEAR produces fidelity statistics, its parameters can be iteratively changed in order to achieve a better trade-off between interpretability and fidelity. Relevant parameters include the number of features/independent variables to consider and the use and number of quadratic or interaction terms. Figure 2 shows excerpts from a CLEAR report.
8. CLEAR also provides the option of adding $\boldsymbol{x}$'s *b*-counterfactuals, $\min_f(\boldsymbol{x})$, to $\boldsymbol{x}$'s neighbourhood dataset. The *b*-counterfactuals are weighted and act as soft constraints on CLEAR's subsequent regression. Algorithms 1 and 2 outline the entire process.

Algorithm 1: CLEAR Algorithm

input : *t* (training data), $\boldsymbol{x}$,*m*,*T*
output: expl (set of explanations)
S← Generate_Synthetic_Data($\boldsymbol{x}$,*t*,*m*)
for *each target class tc* **do**
  for *each feature f* **do**
    $w \leftarrow$ Find_Counterfactuals($\boldsymbol{x}$,*m*)
  end
  $N_{tc} \leftarrow$ Balanced_Neighbourhood(S, $\boldsymbol{x}$, *m*)
  Optional: $N_{tc} \leftarrow N_{tc} \cup w$
  $r \leftarrow$Find_Regression_Equations($N_{tc}, \boldsymbol{x}$)
  $w' \leftarrow$ Estimate_Counterfactuals(r,$\boldsymbol{x}$)
  $e \leftarrow$ Calculate_Fidelity(w,w',*T*)
  return expl$_{tc}$ =$< w, w', r, e >$
end

Algorithm 2: Balanced_Neighbourhood

input : S (synthetic dataset), $\boldsymbol{x}$,*m*
  b_1, b_2 (margins around decision boundary)
output: N (neighbourhood dataset)
$n \leftarrow 200$
for $s_i \in S$ **do**
  $d_i \leftarrow$ Euclidean_Distance $(s_i, \boldsymbol{x})$
  $y_i \leftarrow$m(s_i)
end
$N_1 \leftarrow {}^n/_3$ members of $\{S\}$ with lowest d_i s.t. $0 < y_i \leq b_1$
$N_2 \leftarrow {}^n/_3$ members of $\{S\}$ with lowest d_i s.t. $b_1 < y_i \leq b_2$
$N_3 \leftarrow {}^n/_3$ members of $\{S\}$ with lowest d_i s.t. $b_2 < y_i \leq 1$
return $N \leftarrow N_1 \cup N_2 \cup N_3$

Notice that for CLEAR an explanation (expl) is a tuple $< w, w', r, e >$, where w and w' are *b*-perturbations (actual and estimated), r is a regression equation and e are fidelity errors.

Table 1. Comparison of % fidelity of CLEAR and LIME: the use of a balanced neighbourhood, centering and quadratic terms allow CLEAR, in general, to achieve a considerably higher fidelity to *b*-counterfactuals than LIME, even without training with *b*-counterfactuals. Including training with *b*-counterfactuals (optional step 8 of CLEAR method), % fidelity is further increased.

	Adult	Iris	Pima	Credit	Breast
CLEAR- not using *b*-counterfactuals	80% ± 0.9	80% ± 1.0	57% ± 0.8	39% ± 1.3	54% ± 1.1
CLEAR- using *b*-counterfactuals	80% ± 0.8	99.8% ± 0.1	77% ± 0.8	55% ± 1.7	81% ± 1.3
LIME algorithms	26% ± 0.6	30% ± 0.3	20% ± 0.4	12% ± 0.5	14% ± 0.3

Example of using regression to estimate a *b*-perturbation: An MLP with a softmax activation function in the output layer was trained on a subset of the UCI Pima Indians Diabetes dataset. The MLP calculated a probability of 0.69 for $\boldsymbol{x}$ belonging to class 1 (having diabetes). CLEAR generated the logistic regression equation $(1 + e^{\boldsymbol{w}^T\boldsymbol{x}})^{-1} = 0.69$ where:

$$\boldsymbol{w}^T\boldsymbol{x} = -0.8 + 1.73\ Glucose + 0.25\ BloodPressure + 0.31\ Glucose^2$$

Let the decision boundary be $P(\boldsymbol{x} \in class\ 1) = 0.5$. Thus, $\boldsymbol{x}$ is on the boundary when $\boldsymbol{w}^T\boldsymbol{x} = 0$. The estimated *b*-perturbation for Glucose is obtained by substituting into the regression equation: $\boldsymbol{w}^T\boldsymbol{x} = 0$ and the value of BloodPressure in $\boldsymbol{x}$:

$$-0.31\ Glucose^2 + 1.73\ Glucose - .04 = 0$$

Solving this equation, CLEAR selects the root equal to 0.025 as being closest to the original value of Glucose in $\boldsymbol{x}$. The original value for Glucose was 0.537 and hence the estimated *b*-perturbation is -0.512. The actual *b*-perturbation (step 1) for Glucose to achieve a probability of 0.5 of being in class 1 was -0.557; hence, the fidelity error was 0.045.

A CLEAR prototype has been developed in Python[3]. CLEAR can be run either in batch mode on a test set or it can explain the prediction of a single observation. In batch mode, CLEAR reports the proportion of its estimated *b*-counterfactuals that have a fidelity error lower than a user-specified error threshold *T*, as follows:

*Definition 5.3 (**% fidelity**):* A *b*-perturbation is said to be *feasible* if the resulting feature value is within the range of values found in *m*'s training set. The percentage *fidelity* given a batch and error threshold *T* is the number of *b*-perturbations with fidelity error smaller than *T* divided by the number of feasible *b*-perturbations.

Both fidelity and interpretability are critical for a successful explanation. An 'interpretable explanation' that is of poor fidelity is not an explanation, it is just misinformation. Yet, in order to achieve high levels of fidelity, a CLEAR regression may need to include a large number of terms, including 2nd degree and interaction variables. A criticism of CLEAR might then be that whilst its regression equations are interpretable to data scientists, they may not be interpretable to lay people. However CLEAR's HTML reports are interactive, providing an option to simplify the representation of its equations. Consider Figure 2, if a user was primarily interested in the effects of Glucose and BMI, then CLEAR can substitute the values of the other features leading to the full regression equation being re-expressed as:

$$\boldsymbol{w}^T\boldsymbol{x} = 0.46 + 0.3\ Glucose + 0.18\ BMI - 0.05\ BMI^2$$

This equation helps to explain the *b*-counterfactuals, for example it shows why observation 1's classification is more easily 'flipped' by changing the value of Glucose rather than BMI. CLEAR additionally provides options for the user to simplify the original regression, for example by reducing the number of terms, excluding interaction terms etc. CLEAR then enables the user to see the resulting fall in fidelity, putting the user in control of the interpretability/fidelity trade-off. Figure 3 illustrates how fidelity is reduced by excluding both quadratic and interaction terms; notice that even both are excluded, CLEAR's fidelity is much higher than LIME's.

[3] https://github.com/ClearExplanationsAI

4 Experimental Results

Experiments were carried out with five UCI datasets: Pima Indians Diabetes (with 8 numeric features), Iris (4 numeric features), Default of Credit Card Clients (20 numeric features, 3 categorical), and subsets of Adult (with 2 numeric features, 5 categorical features), and Breast Cancer Wisconsin (9 numeric features). For the Adult dataset, some of the categorical features values were merged and features with little predictive power were removed. With the Breast Cancer dataset only the mean value features were kept. For reproducibility, the code for pre-processing the data is included with the CLEAR prototype on GitHub.

For the Iris dataset, a support vector machine (SVM) with RBF kernel was trained using the scikit-learn library. For each of the other datasets, an MLP with a softmax output layer was trained using Tensorflow. Each dataset was partitioned into a training dataset (used to train the SVM or MLP, and to set the parameters used to generate synthetic data for CLEAR) and a test dataset (out of which 100 observations were selected for calculating the % fidelity of CLEAR and LIME). Experiments were carried out with different test sets, with each experiment being repeated 20 times for different generated synthetic data. The experiments were carried out on a Windows i7-8700 3.2GHz PC. A single run of a 100 observations took 40-80 minutes, depending on the dataset.

In order to enable comparisons with LIME, CLEAR includes an option to run the LIME algorithms for creating synthetic data and generating regression equations. CLEAR then takes the regression equations and calculates the corresponding *b*-counterfactuals and fidelity errors. It might be objected that it is unfair to evaluate LIME on its counterfactual fidelity, as it was not created for this purpose. But if you accept Woodward and Miller's requirements for a satisfactory explanation, then counterfactual fidelity is an appropriate metric.

CLEAR's regressions are significantly better than LIME's. The best results are obtained by including *b*-counterfactuals in the neighbourhood datasets (step 8 of the CLEAR method). Overall, the *best* configuration comprised: using balanced neighbourhood data, forcing the regression curve to go through $\boldsymbol{x}$ (i.e. 'centering'), including both quadratic and interaction terms, and using logistic regression for Pima and Breast Cancer and multiple regression for Iris, Adult and Credit datasets. Unless otherwise stated % fidelity is for the error threshold *T* = 0.25. Table 1 compares the % fidelity of CLEAR and LIME (i.e. using LIME's algorithms for generating synthetic data

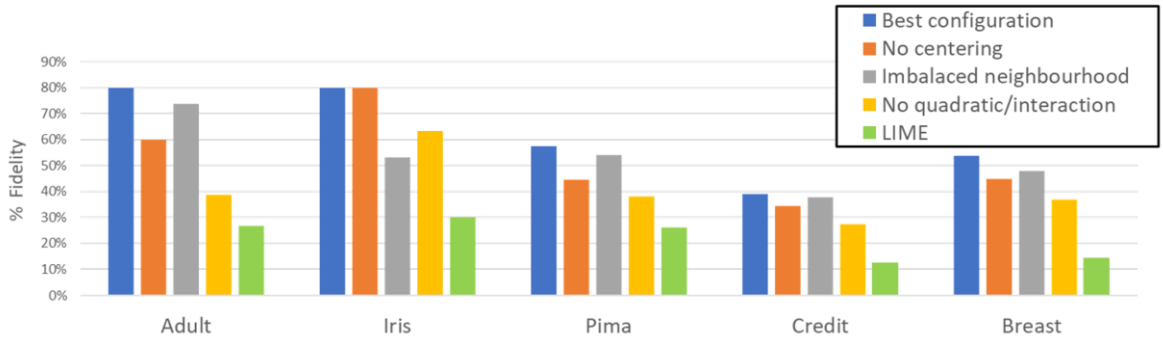

Figure 3. Comparison of fidelity with different configurations. Fidelity is reduced when any of the above changes are made to the *best configuration*. The changes include:'Imbalanced neighbourhood' = 'using best configuration but with imbalanced neighbourhood datasets', and 'LIME' = 'using the LIME algorithms'. Notice the benefits of CLEAR using a balanced neighbourhood dataset. Also, when CLEAR has 'no quadratic and interaction terms', it is still significantly more faithful than LIME.

and performing the regression). This used LIME's default parameter values except for the following beneficial changes: the number of synthetic data points was increased to 15,000 (further increases did not improve fidelity), the data was not discretized, a maximum of 14 features were allowed, several kernel widths in the range from 1.5 to 4 were evaluated. By contrast, CLEAR was run with its *best* configuration and with 14 features. As an example of LIME's performance: with the Credit dataset, the adjusted R^2 averaged ≈ 0.7, the classification of the test set observations was over 94% correct. However, the absolute error between y and LIME's estimate of y was 8% (e.g. the MLP forecast $P(\boldsymbol{x} \in class\ 1) = 0.4$, while LIME estimated it at 0.48) and this by itself would lead to large errors when calculating how much a single feature needs to change for y to reach the decision boundary. LIME's fidelity of only 12%, illustrates that CLEAR's measure of fidelity is far more demanding than just classification accuracy. Of course, LIME's poor fidelity was due, in part, to its kernel failing to isolate the appropriate neighbourhood datasets necessary for calculating b-counterfactuals accurately.

Table 2 shows how CLEAR's fidelity (not using b-counterfactuals) varied with the maximum 'number of independent variables' allowed in a regression. At first, fidelity sharply improves but then plateaus.

Despite CLEAR's regression fitting the neighbourhood data well, a significant number of the estimated b-counterfactuals have large fidelity errors. For example, in one of the experiments with the Adult dataset where the multiple regression did not center the data, the average adjusted R^2 was 0.97, classification accuracy 98% but the % fidelity error < 0.25 was 59%. This points to a more general problem: sometimes the neighbourhood datasets do not represent the regions of its feature space that are central for its explanations. With CLEAR, this discrepancy can at least be measured.

Table 2. Variation of % fidelity with the choice of number of variables

No.	Adult	Iris	PIMA	Credit	Breast
8	35%	50%	42%	27%	43%
11	76%	62%	53%	38%	46%
14	80%	80%	57%	39%	54%
17	78%	n/a	59%	40%	55%
20	78%	n/a	62%	39%	56%

CLEAR was tested in a variety of configurations. These included the *best* configuration, and configurations where a single option was altered from the default, e.g. by using a imbalanced neighbourhood of points nearest to $\boldsymbol{x}$. Figure 3 displays the results when CLEAR used a maximum of 14 independent variables.

CLEAR's fidelity was sharply improved by adding $\boldsymbol{x}$'s b-counterfactuals to its neighbourhood datasets. In the previous experiments, CLEAR created a neighbourhood dataset of at least 200 synthetic observations, each being given a weighting of 1. This was now altered so that each b-counterfactual identified in step 1 was added and given a weighting of 10. For example, for the Pima dataset, an average of $\approx$3 b-counterfactuals were added to each neighbourhood dataset. The consequent improvement in fidelity indicates as expected that adding these weighted data points results in a dataset capable of representing better the relevant neighbourhood, with CLEAR being able to provide a regression equation that is more faithful to b-counterfactuals.

5 Conclusion and Future Work

CLEAR satisfies the requirement that satisfactory local explanations should include statements of key counterfactual cases. CLEAR explains a prediction y for data point $\boldsymbol{x}$ by stating $\boldsymbol{x}$'s b-counterfactuals and providing a regression equation. The regression shows the patterns of counterfactual dependencies in a neighbourhood that includes both $\boldsymbol{x}$ and the b-counterfactual data points. CLEAR represents a significant improvement both on LIME and on just using b-counterfactuals. Crucial to CLEAR's performance is the ability to generate relevant neighbourhood data bounded by its b-counterfactuals. Adding these to $\boldsymbol{x}$'s neighbourhood led to sharp fidelity improvements. Another key feature of CLEAR is that it reports on its own fidelity. Any local explanation system will be fallible, and it is critical with high-value decisions that the user knows if they can trust an explanation.

There is considerable scope for further developing CLEAR. These include (i) the neighbourhood selection algorithms could be further enhanced. Other data points could also be added, for example b-counterfactuals involving changes to multiple features. A user might also include some perturbations that are important to their project. CLEAR could guide this process by reporting regression and fidelity statistics. And step 8 of the CLEAR algorithm could be replaced by increasingly complex and more sophisticated learning algorithms. Constructing neighbourhood datasets in this way would seem a better approach than randomly generating data points and then selecting those closest to $\boldsymbol{x}$ (ii) the search algorithm for actual b-perturbations in step 1 of the CLEAR algorithm could be replaced by a more computationally efficient algorithm (iii). CLEAR should be evaluated in practice in the context of comprehensibility studies.

REFERENCES

[1] Robert Andrews, Joachim Diederich, and Alan B. Tickle, 'Survey and critique of techniques for extracting rules from trained artificial neural networks', *Knowledge-Based Systems*, **8**(6), 373–389, (December 1995).

[2] Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin, 'Learning certifiably optimal rule lists for categorical data', *Journal of Machine Learning Research*, **18**, 234:1–234:78, (2017).

[3] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba, 'Network dissection: Quantifying interpretability of deep visual representations', *In Proc. CVPR 2017, Honolulu, USA*, (2017).

[4] Mark W. Craven and Jude W. Shavlik, 'Extracting tree-structured representations of trained networks', in *Proc. NIPS 1995*, pp. 24–30. MIT Press, (1995).

[5] Zoubin Ghahramani, 'Probabilistic machine learning and artificial intelligence', *Nature*, **521**(7553), 452–459, (2015).

[6] Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti, 'A survey of methods for explaining black box models', *CoRR*, **abs/1802.01933**, (2018).

[7] Patrick Hall, Navdeep Gill, Megan Kurka, and Wen Phan. Machine learning interpretability with H2O driverless AI, 2017. URL: http://docs.h2o.ai/ driverless-ai/latest-stable/docs/booklets/MLIBooklet.pdf.

[8] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner, 'beta-vae: Learning basic visual concepts with a constrained variational framework.', *ICLR*, **2**(5), 6, (2017).

[9] Linwei Hu, Jie Chen, Vijayan N. Nair, and Agus Sudjianto, 'Locally interpretable models and effects based on supervised partitioning (LIME-SUP)', *CoRR*, **abs/1806.00663**, (2018).

[10] Henrik Jacobsson, 'Rule extraction from recurrent neural networks: A taxonomy and review', *Neural Computation*, **17**(6), 1223–1263, (June 2005).

[11] Scott M Lundberg and Su-In Lee, 'A unified approach to interpreting model predictions', in *Advances in Neural Information Processing Systems*, pp. 4765–4774, (2017).

[12] Tim Miller, 'Contrastive explanation: A structural-model approach', *arXiv preprint arXiv:1811.03163*, (2018).

[13] Tim Miller, 'Explanation in artificial intelligence: Insights from the social sciences', *Artificial Intelligence*, (2018).

[14] Saumitra Mishra, Bob L Sturm, and Simon Dixon, 'Local interpretable model-agnostic explanations for music content analysis.', in *ISMIR*, pp. 537–543, (2017).

[15] Brent Mittelstadt, Chris Russell, and Sandra Wachter, 'Explaining explanations in AI', in *In Proc. Conference on Fairness, Accountability, and Transparency*, FAT'19, pp. 279–288, New York, USA, (2019).

[16] Judea Pearl, *Causality: Models, Reasoning and Inference*, Cambridge University Press, New York, NY, USA, 2nd edn., 2009.

[17] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, 'Why should i trust you? explaining the predictions of any classifier', in *Proc. ACM SIGKDD 2016*, KDD '16, pp. 1135–1144, New York, NY, USA, (2016). ACM.

[18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, 'Anchors: High-precision model-agnostic explanations', in *Thirty-Second AAAI Conference on Artificial Intelligence*, (2018).

[19] Son N. Tran and Artur S. d'Avila Garcez, 'Deep logic networks: Inserting and extracting knowledge from deep belief networks', *IEEE Transactions on Neural Networks and Learning Systems*, **29**(2), 246–258, (2018).

[20] Sandra Wachter, Brent D. Mittelstadt, and Chris Russell, 'Counterfactual explanations without opening the black box: Automated decisions and the GDPR', *CoRR*, **abs/1711.00399**, (2017).

[21] Qinglong Wang, Kaixuan Zhang, Alexander G. Ororbia, II, Xinyu Xing, Xue Liu, and C. Lee Giles, 'An empirical evaluation of rule extraction from recurrent neural networks', *Neural Computation*, **30**(9), 2568–2591, (September 2018).

[22] J. Woodward, *Making things happen: a theory of causal explanation*, Oxford University Press, Oxford, England, 2003.