FIMAP: Feature Importance by Minimal Adversarial Perturbation

Matt Chapman-Rounds[1], **Umang Bhatt**[2], **Erik Pazos**[3], **Marc-Andre Schulz**[4], **Konstantinos Georgatzis**[3]

[1]Department of Informatics, University of Edinburgh,
[2]Department of Engineering, University of Cambridge,
[3]QuantumBlack,
[4]Department of Psychiatry and Psychotherapy, Charité–Universitätsmedizin Berlin
m.rounds@ed.ac.uk

Abstract

Instance-based model-agnostic feature importance explanations (LIME, SHAP, L2X) are a popular form of algorithmic transparency. These methods generally return either a *weighting* or *subset* of input features as an explanation for the classification of an instance. An alternative literature argues instead that *counterfactual* instances, which alter the black-box model's classification, provide a more actionable form of explanation. We present Feature Importance by Minimal Adversarial Perturbation (FIMAP), a neural network based approach that unifies feature importance and counterfactual explanations. We show that this approach combines the two paradigms, recovering the output of feature-weighting methods in continuous feature spaces, whilst indicating the direction in which the nearest counterfactuals can be found. Our method provides an implicit confidence estimate in its own explanations, something existing methods lack. Additionally, FIMAP improves upon the speed of sampling-based methods, such as LIME, by an order of magnitude, allowing for explanation deployment in time-critical applications. We extend our approach to categorical features using a partitioned Gumbel layer and demonstrate its efficacy on standard datasets.

Introduction

Recent interest in explaining the output of complex machine learning (ML) models has been characterized by a wide range of approaches (Lipton 2016; Montavon, Samek, and Müller 2018). Many of these approaches are model specific; for example, attempts to explain neural networks rely on interpreting the flow of gradients through the model (Karpathy, Johnson, and Fei-Fei 2015; Shrikumar, Greenside, and Kundaje 2017; Olah, Mordvintsev, and Schubert 2017), or decision trees, which might be considered directly interpretable, provide explanations as rules (Molnar 2019).

Model agnostic approaches, however, are attempts to formulate a general framework for per-instance explanation of a model's outputs regardless of the model class. This can be beneficial when the choice of model may change over time or where the original model is costly to query.

One group of model-agnostic explainers focuses on providing an explanation of a model's output as either a subset of input features (Ribeiro, Singh, and Guestrin 2018; Chen et al. 2018) or a weighting of input features (Ribeiro, Singh, and Guestrin 2016; Lundberg and Lee 2017) of the instance to be explained. Another set of methods proposes that counterfactual instances, or groups of instances, are a useful proxy to 'explanation,' where the claim is that local explanations are expected to contain both the outcome of a prediction and how that prediction would change if the input changed (Wachter, Mittelstadt, and Russell 2017; White and Garcez 2019). Many such approaches use sampling procedures to either estimate local decision boundaries (and their corresponding parameters) or to find proximate counterfactual instances: in both cases, they can be computationally expensive (Bhatt, Weller, and Moura 2020). The computational cost of sampling local decision boundaries for each explanation makes these methods slow to scale and of limited use in practice (Bhatt et al. 2019).

Herein, we propose FIMAP, Feature Importance by Minimal Adversarial Perturbation, a model that returns the direction that an instance would have to be perturbed the least in order for the classification of the underlying model to change. FIMAP's contribution is threefold:

- FIMAP combines elements of both feature importance and counterfactual explanations, and is model-agnostic.
- FIMAP is faster than alternative methods by 5 orders of magnitude, once constant overheads are taken into account, allowing for model explanations in time-critical applications where sampling-based methods are infeasible.
- FIMAP naturally indicates regions of low classifier confidence as a consequence of its design.

The paper is structured as follows. We first provide an summary of recent approaches to instance-wise model-agnostic explanation. We then show how our method unifies counterfactual and feature importance explanations, justify FIMAP's approach, and describe how we handle continuous and categorical input variables. Before concluding, we show empirical results on synthetic and real-world experiments.

Related Work

One of the most widely-used feature weighting approaches to per-instance explanation of a black box model's outputs is LIME (Ribeiro, Singh, and Guestrin 2016), which learns a local surrogate approximation to the black box model's

output centered on the instance to be explained. It first generates a new dataset of permuted samples and corresponding predictions of the black box model, and then trains an interpretable linear model on this new dataset, where each point is weighted by its proximity to the point of interest. The weights of the linear model are then considered to be the explanations of the black box model's output at that point. LIME can also be considered to be slow; its reliance on sampling afresh for every data point reduces the speed at which explanations can be collected for large numbers of instances.

Separate work has shown that those explanation methods that return a weighting of input features, including LIME, can all be considered as additive feature attribution methods, with an explanation model that is a linear function of binary variables (Lundberg and Lee 2017). This unified framework is called SHAP, and accompanying methods exist to estimate feature importance values for instance predictions on particular models (Lundberg, Erion, and Lee 2018).

One attempt to produce *fast* instance-based explanations is L2X (Chen et al. 2018), where the authors train a neural network to output a binary mask over instance features, and a second network to return the original black box model output from the masked input. By training on a cross entropy objective, they argue that they are effectively maximising the mutual information between some subset of input features and the true model output. The subset of features chosen once the explainer is trained should be the maximally informative subset, and thus a good explanation of the black box model output. This approach shares some similarity with ours, insofar as the second network can be thought of as learning a differentiable surrogate to the true model, although the authors do not consider their model in these terms. A crucial drawback of L2X is that it does not provide weighting of feature importances, nor does it provide the direction in which a given feature would impact classification.

An example of the fact that adversarial examples can be good explanations of underlying models is the work of Wachter, Mittelstadt, and Russell. Here the approach, assuming a trained model $f_w(x)$, is to minimise

$$L(x, x', y', \lambda) = \lambda(f_w(x') - y')^2 + d(x, x') \, ,$$

where the first term is the quadratic distance between the output of the model under some counterfactual input x' and a new target y', and the second term is a measure of the distance between the true input to be explained, x, and its possible counterfactual instance x'. This approach is similar in spirit to ours, but differs in several important ways.

Firstly the method returns a set of counterfactual instances, rather than a counterfactual direction. Secondly, the procedure to generate one counterfactual example for one point requires iterating between minimising the above objective and increasing λ, and the authors recommend initialising a sample of potential counterfactuals and repeating the process on all of them, to avoid getting stuck in local minima. This means the process is slow. Thirdly, optimising the above objective assumes that $f_w(x)$ is tractable (for example, a gradient based optimiser would need the gradient of $f_w(x)$ with respect to x). This limits the approach to only those models where this is the case, whereas by training a differentiable approximation to the black box model, we circumvent this issue. Another similar approach can be found in CLEAR, (White and Garcez 2019), which includes an interesting model of fidelity, although again the process of extracting an explanation requires sampling, and iterative solving. In short, LIME, SHAP, and other sampling-based models require thousands of model-evaluations for each instance that needs to be explained. L2X needs only one forward pass of a neural network per explanation, but does not provide a weighting of feature importances, nor directionality of explanations. With FIMAP, we provide a method that retains the benefits of LIME and SHAP, while providing computational efficiency on par with L2X.

When explaining the outputs of neural networks, particularly for image classification, there are several examples of papers which use adversarial or perturbation approaches (Dabkowski and Gal 2017; Dhurandhar et al. 2018; Fong and Vedaldi 2017; Zhao, Dua, and Singh 2017; Cheng et al. 2018). These approaches often rely on dividing images into regions, which places a strong modelling prior on correlations between input features (here, pixels). As our approach is fundamentally more general, we are not able to make similar assumptions, and likely would have substantially different use-cases. Two such papers (Dabkowski and Gal 2017; Dhurandhar et al. 2018) assume differentiability, whilst the third treats 'perturbations' as a regional noise masks; instead of learning feature-specific meaningful perturbations as in our approach. Dabkowski and Gal use GANs to generate 'natural' adversarial examples, by finding adversaries which are similar but also interpretable. This approach is useful in domains where individual features are not necessarily meaningful (such as images), but may occlude specific feature importance in the name of 'naturalness'. (Cheng et al. 2018) also provide a black box approach to finding adversarial examples, but their search requires thousands of queries per image, rendering them far slower than our approach.

Model

Overview

Our general approach to the problem of explaining an instance's classification by a model is to find the minimal *adversarial* perturbation of that instance. This can be thought of as an answer to the question 'what is the smallest change we can make to this instance to change its classification?'. We argue that this is a useful measure for two reasons.

First, it is locally meaningful. An instance's classification depends on its location relative to the classifier's decision boundary or boundaries. The minimal adversarial perturbation will 'point' directly to the nearest decision boundary. Features that contribute substantially to this minimal perturbation must also be features that have contributed substantially to the instance's classification. If we imagine perturbing the features of an instance equally, those with relatively large contributions to the original classification will be just those that have a relatively large contribution to subsequent misclassification.

Secondly, it is useful for an end-user. The outputs of a model often require explanation due to a desire for improve-

ment, or, more specifically, instances that require further justification are often instances which have been wrongly classified, or are suspected to have been wrongly classified. Indicating what should be changed to allow an instance to be alternatively classified satisfies this requirement directly, and in a manner which is arguably more interpretable than providing the weights of a local linear model.

Continuous Input Features

Let us assume we have access to a set of outputs $\{f(\boldsymbol{x}^{(n)})\}_{n=1}^{N}$ of some model $f : \mathbb{R}^d \rightarrow \mathbb{R}$, where for binary classification, $f(\boldsymbol{x}^{(n)})$ will be the probability that input instance $\boldsymbol{x}^{(n)}$ belongs to the target class [1], or a corresponding indicator function ($\mathbb{1}[f(\boldsymbol{x}^{(n)})] = 1, f(\boldsymbol{x}^{(n)}) \geq 0.5$). For each $\boldsymbol{x}^{(n)}$ we wish to explain, our goal is to find the smallest adversarial perturbation; i.e. the smallest perturbation $\boldsymbol{p}^{(n)}$ such that $\mathbb{1}[f(\boldsymbol{x}^{(n)})] = 1 - \mathbb{1}[f(\boldsymbol{x}^{(n)} + \boldsymbol{p}^{(n)})]$. Here, $\boldsymbol{p}^{(n)} \in \mathbb{R}^d$, and if minimal, can be thought of as the shortest distance from $\boldsymbol{x}^{(n)}$ to the decision boundary of f.

The space of possible perturbations P is prohibitively large for an exhaustive search per instance to be explained, and so we will assume a restricted class of models $G : X \rightarrow P$, mapping data space to perturbations. Our approach in this paper is to represent such a mapping as $g(\boldsymbol{x}; \theta_g) : \mathbb{R}^d \rightarrow \mathbb{R}^d$, $g \in G$, a differentiable function described by a neural network with parameters θ_g. Ideally, we would then like to compute the optimal adversarial parameter settings $\hat{\theta}_g$ by standard gradient-based methods, using:

$$\hat{\theta}_g = \underset{\theta_g}{\operatorname{argmin}} \Big\{ -\sum_{n=1}^{N} \Big(1 - \mathbb{1}[f(\boldsymbol{x}^{(n)})]\Big) \log f\Big(\boldsymbol{x}^{(n)} + g(\boldsymbol{x}^{(n)}; \theta_g)\Big) + \lambda |g(\boldsymbol{x}^{(n)}; \theta_g)|_2 \Big\} , \tag{1}$$

where λ is a hyperparameter restricting the size of generated perturbations, and $1 - \mathbb{1}[f(\boldsymbol{x}^{(n)})]$ are the adversarial labels.

However, in a model-agnostic setting, we cannot assume f to be differentiable[2], or even that we have access to f itself to compute $f(\boldsymbol{x}^{(n)} + g(\boldsymbol{x}^{(n)}; \theta_g))$. We therefore further define a surrogate $s(\boldsymbol{x}; \theta_s) : \mathbb{R}^d \rightarrow \mathbb{R}$, also a neural network, which is trained to be a differentiable approximation to f by cross entropy loss:

$$\hat{\theta}_s = \underset{\theta_s}{\operatorname{argmin}} - \sum_{n=1}^{N} \mathbb{1}[f(\boldsymbol{x}^{(n)})] \log s(\boldsymbol{x}^{(n)}; \theta_s) . \tag{2}$$

Substituting $s(x; \hat{\theta}_s)$ for f in (1) finally gives us a tractable objective:

$$\hat{\theta}_g = \underset{\theta_g}{\operatorname{argmin}} \Big\{ -\sum_{n=1}^{N} \Big(1 - \mathbb{1}[f(\boldsymbol{x}^{(n)})]\Big) \log s\Big(\boldsymbol{x}^{(n)} + g(\boldsymbol{x}^{(n)}; \theta_g); \hat{\theta}_s\Big) + \lambda |g(\boldsymbol{x}^{(n)}; \theta_g)|_2 \Big\} . \tag{3}$$

Note that $\mathbb{1}[f(\boldsymbol{x}^{(n)})]$ remains unchanged, as it does not depend on θ_g, and we have assumed we know $f(\boldsymbol{x}^{(n)})$ for all $\boldsymbol{x}^{(n)}$ in our data.

In practice, training is carried out in two stages; firstly we train $s(\boldsymbol{x}; \theta_s)$ on the original inputs and original labels to approximate the black box model f. Secondly, we freeze the weights of s and train $g(\boldsymbol{x}; \theta_g)$ on the original inputs and flipped labels; the perturbations $\boldsymbol{p}^{(n)}$ output by $g(\boldsymbol{x}^{(n)}; \theta_g)$ are added to the original inputs and passed through the surrogate s. As s is a differentiable model, back-propagation provides the gradients of the loss with respect to the perturbations, and hence with respect to θ_g. We can therefore train g directly using the original dataset.

Discrete Input Features

For many applications, however, some or all of the input features of f will be discrete, rather than continuous. For some categorical feature x_i, which takes values $\{1, ..., K\}$ outputting a continuous value p_i from our perturbation generator g is unhelpful. We first consider the case in which all input features are categorical.

One approach, if we have access to a meaningful embedding, would be to perturb the real-valued representations of each categorical feature within the embedding space. However, whilst this would provide us with an indication of the direction in embedding space each input feature should be moved to force the underlying model f to miss-classify, we would have to provide further post-hoc analysis to explain to the user what a move in an abstract embedding space means in terms of the real categorical feature.

We take the general approach that perturbing a categorical feature means sampling from a corresponding categorical distribution and assigning the feature the sampled value. For each categorical x_i, our mapping g from data space to perturbation space contains the corresponding sub-mapping $g_i(x_i^{(n)}, \theta_g) : \mathbb{R}^K \rightarrow \mathbb{R}^K$, assuming a 1-hot encoding, where each of the K real valued outputs is treated as the log class probability $\log \pi_k$ of the k^{th} value of the categorical feature.

To train to find adversarial samples, we can use the softmax function as a continuous differentiable approximation to $argmax$, which allows us to use the Gumbel-Softmax trick to generate K-dimensional sample vectors y where the k^{th} element is given by:

$$y_k = \frac{\exp((\log \pi_k + g_k)/\tau)}{\sum_{j=1}^{K} \exp((\log \pi_j + g_j)/\tau)} , \tag{4}$$

where τ is a hyperparameter governing the temperature of the distribution; as it approaches 0, the Gumbel-Softmax distribution approaches the Categorical distribution. $g_k \sim -\log(-\log(U))$, where $U \sim$ UNIFORM$(0, 1)$. This path

[1] For the sake of clarity, we will initially assume a binary classification. Multi-class classification is dealt with below, and regression is discussed in the conclusion

[2] Or at least, we cannot assume we have access to the gradients of f.

derivative estimator allows us to backpropagate through the parameters of the sample for each categorical variable and thus train the perturbation model g (see Jang, Gu, and Poole for more details).

When training g, these samples are then concatenated into a perturbed instance, $\boldsymbol{p}^{(n)}$, which is passed through the pre-trained surrogate model $s(\boldsymbol{p}^{(n)}; \hat{\theta}_s)$ as before.

The only other difference to the training procedure is that the term in the objective intended to minimise the size of the adversarial perturbations in (3), $\lambda|g(\boldsymbol{x}^{(n)}; \theta_g)|_2$, must be changed to account for the fact we are no longer perturbing by adding small vectors to an input in $\mathbb{R}^d$. We make the simplest assumption that if perturbed feature $p_i^{(n)}$ takes on the same value as the original feature $x_i^{(n)}$, it has a perturbation cost of 0, and otherwise has a cost proportional to a hyper-parameter η. This yields the following regularisation term:

$$\text{reg}(\boldsymbol{x}^{(n)}) = \eta \sum_i^D \frac{1}{2} \left| x_i^{(n)} - p_i^{(n)} \right| \tag{5}$$

Where $x_i^{(n)}$ and $p_i^{(n)}$ are 1-hot vectors of length K (which may be different for different i), and D is the number of categorical variables in $\boldsymbol{x}$.

Our approach also supports a hybrid of both categorical and continuous variables, by combining the two objectives outlined above, where each affects the appropriate variables. The main challenge here is the relative magnitudes of λ and η. We found (see discussion in Results, below), that for simple datasets setting λ to around an order of magnitude smaller than η yielded good results.

Connecting Counterfactual Explanations and Feature Importance

Counterfactual explanations can be seen as adversarial examples with feasibility and plausibility constraints. Recent works has explored the intersection between adversarial robustness, counterfactual explanations, and feature importance (Yeh et al. 2019; Etmann et al. 2019; Singla et al. 2019; Ghorbani, Abid, and Zou 2019; Dombrowski et al. 2019).

Here, we take the minimum adversarial perturbation to be a feature importance explanation. To encourage robustness, Singla et al. solve an objective that maximises the log likelihood subject to minimising the top k feature importance score of the data, x.

This objective is similar to the counterfactual explanation objective except the latter would contain an additional constraint to flip the predicted class label $\vec{f}_\theta(\tilde{\vec{x}}) \neq \vec{f}_\theta(\vec{x})$ and may not include the ℓ_0 norm to limit the number of features that changed; though, Su, Vargas, and Sakurai successfully perform an adversarial attack by perturbing only one feature, $k = 1$. This connection indicates that feature importance can be cast as a specific case of a counterfactual explanation without plausibility, feasibility, or class constraints (Sharma, Henderson, and Ghosh 2019).

How does this align with our model? Consider Equation (3) if $s(x; \hat{\theta}_s)$ were a simple linear model such as logistic regression, where $s(x; \hat{\theta}_s) = 1/(1 + \exp(-x^T \hat{\theta}_s))$,

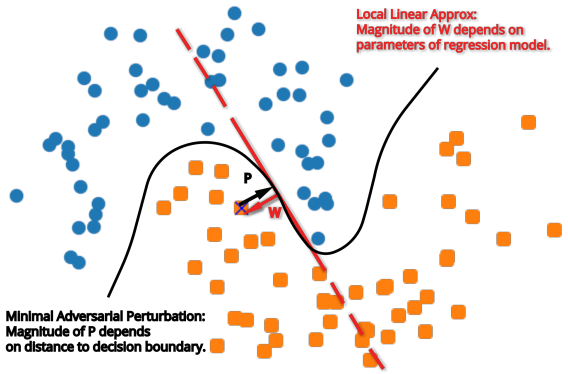

Figure 1: A pictorial representation of the intuition that the minimal adversarial perturbation should recover the feature importance of a local linear approximation to the decision boundary. The magnitudes of P (the output of $g(x)$) and W (in text, $\hat{\theta}_s$) may not be the same.

and the data were such that a linear model could achieve good accuracy. The trained model would describe a decision plane through data space where $\hat{\theta}_s$ would be a vector perpendicular to this plane, pointing in the direction for which $\mathbb{1}[f(\boldsymbol{x}^{(n)})] = 1$. It should be clear that in this case a successful adversarial training process will learn an output vector $g(x; \hat{\theta}_g)$ which may differ in magnitude from $\hat{\theta}_s$ but will be anti-parallel to it [3].

LIME relies on the assumption that locally data are always such that a linear model could achieve good accuracy. If the data we were training on above was local to a point of interest, our linear $s(x)$ and $g(x)$ would recover the negative relative feature importance a la LIME (potentially with a different magnitude). In the general case, where $s(x)$ and $g(x)$ are much more expressive, the regularisation term $\lambda|g(\boldsymbol{x}^{(n)}; \theta_g)|_2$ constrains the outputs of $g(x)$ to be effectively local. By minimising the magnitude of the perturbation, we force it to be perpendicular to the tangent to the decision boundary at the point closest to the point of interest. If the decision boundary can be reasonably approximated as piece-wise linear, then the perturbation will recover the feature importance naturally (see Figure 1).

Experimental Setup

For all experiments below, expect otherwise stated, the neural network parameterising $g(\boldsymbol{x}; \theta_g)$ consists of four fully connected layers of size 100 with ReLU nonlinearities and a 'partial gumbel layer' that combines standard additive perturbations for continuous variables with a collection of

[3] To visualise this, consider the case where $g(x)$ is a single layer neural network (linear regression). For each dimension d, training on the reverse labels will learn a straight line the gradient and intersect of which will be determined by the relative distribution (in that dimension) of $\mathbb{1}[f(\boldsymbol{x}^{(n)})] = 1$ versus $\mathbb{1}[f(\boldsymbol{x}^{(n)})] = 0$. The combination of these regression values will be a vector in data space which will point towards the inferred decision boundary.

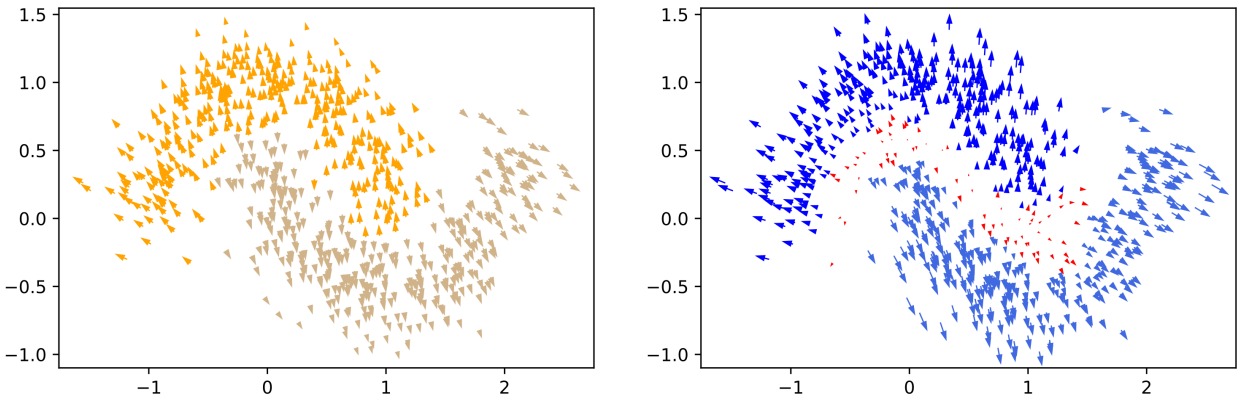

Figure 2: Performance of instance-wise explainers on half moons data. Note that for the vast majority of points, LIME (orange) and FIMAP (blue) provide near identical explanations (indicated by the direction of the arrows). Axis are arbitrary features x_1 and x_2. (Left): LIME x_1, x_2 coefficients plotted as vectors starting at the location of the point to be explained. (Right): FIMAP negative perturbations plotted as vectors starting at the location of the point to be explained. Smallest 10% of FIMAP vectors indicated in red.

Gumbel-Softmax outputs for categorical variables, as discussed in the 'Model' section, above. We used a dropout percentage of 20 for every layer.

The neural network parameterising the surrogate $s(\boldsymbol{x}; \theta_s)$ consists of three fully connected layers of size 200, with the first two nonlinearities being ReLU, and the final Softmax. We used a cross-entropy loss, as is standard for classification, and trained both models using Adam (Kingma and Ba 2014), with a learning rate of 1e-3. On simple synthetic datsets, both networks converge in under 15 epochs.

Network architecture and hyperparameters were chosen to be simple whilst providing reasonable results on a variety of datasets. Our intention was to showcase the generality and robustness of our model, so we avoided hyperparameter tuning or intensive model selection. Several similar architectures (more layers, wider fully connected layers) worked equally well, and an analysis of their relative merits is not pertinent to this initial presentation of the model.

For simple synthetic data we used 10000 samples from the half moons dataset, available on scikit-learn (Pedregosa et al. 2011), with Gaussian noise with standard deviation 0.2 added to the data. Our second synthetic dataset was handcrafted, and is described in the Results section, below. A more realistic continuous dataset was MNIST (LeCun et al. 1998), which we converted to a binary classification task by using only the digits 8 and 3, which gave a train/test split of 11982/1984, and training a classifier to predict between them. This approach was followed by both Lundberg and Lee and Chen et al..

Finally, to test the performance of our method on a mix of categorical data and continuous data, we used a dataset available from the UCI machine learning repository (Dua and Graff 2017). This was a subset of the Adult dataset, where in a similar fashion to White and Garcez, uninformative or highly skewed features ('fnlwgt', 'education', 'relationship', 'native-country', 'capital-gain', 'capital-loss') were removed, along with instances with missing values. The two classes were then balanced by undersampling the larger class, yielding a 17133/5711 train/test split. This left 3 continuous features, which were normalised to have zero mean and unit variance, and 5 categorical features (see Table 1 for example instances).

Results

We first demonstrate that in simple continuous input spaces, FIMAP closely approximates LIME on standard a synthetic dataset, and succeeds in highlighting regions of interest in a manner unavailable to LIME.

We trained a Random Forest with 200 trees to classify the half-moons dataset (with a train/test split of 8000/2000) provided as standard with the scikit-learn toolset (Pedregosa et al. 2011). The classifier had an f-score of 0.97 on the test set. We then generated explanation coefficients for the classification 2000 randomly sampled points in the dataset using the off-the-shelf LIME toolkit (Ribeiro, Singh, and Guestrin 2016). Figure 2 (left) shows 750 of these coefficients plotted as vectors starting at the location of the point to be explained.

Next, we trained our surrogate on the input/output pairs of the Random Forest classifier, again with a 8000/2000 split. A surrogate achieved a recovery accuracy of 0.981 on the test set[4]. We then trained our perturbation network on the opposite class labels, and it achieved an adversarial accuracy of 0.977 on the test set. Figure 2 (right) shows the negative minimal perturbations returned by the perturbation network for the same 750 points explained by LIME.

We present the negative perturbations for ease of comparison - by construction, minimal perturbations will point towards the nearest decision boundary whilst the weights of LIME's fitted logistic regression will point away from the nearest decision boundary. If presenting FIMAP's outputs as explanations of the actual classification *a la* LIME, this

[4]It recovered the Random Forest's classification 98% of the time.

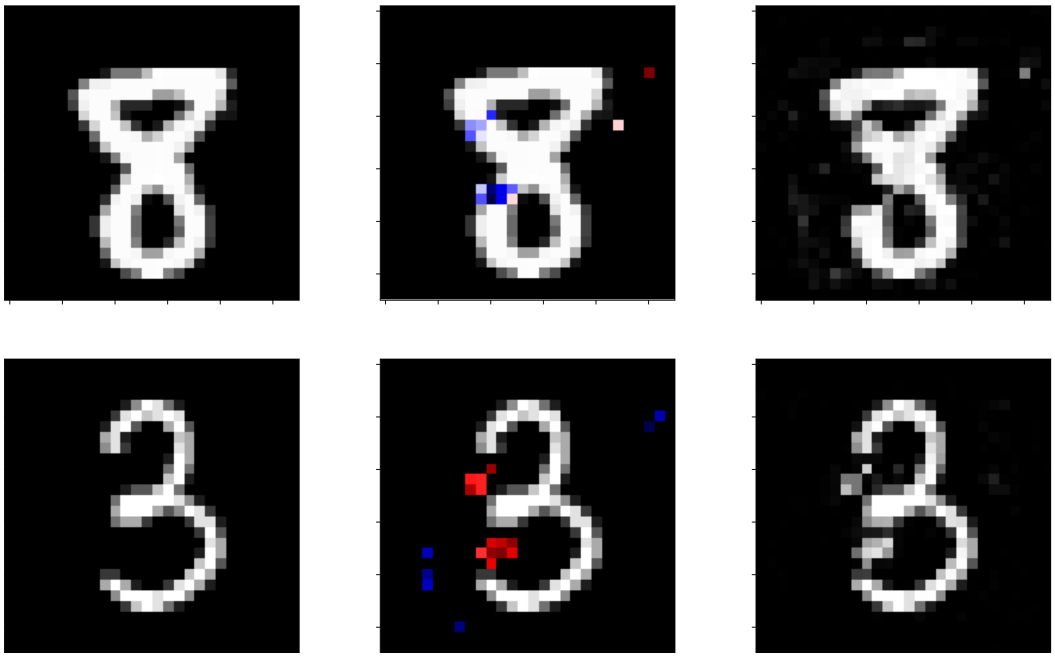

Figure 3: FIMAP perturbations to MNIST digits at the pixel level. Left column is the original digit, middle denotes the FIMAP perturbation, and the right column shows the perturbed digit. (Top): minimal perturbation to flip the Random Forest's classification from '8' to '3', and surrogate softmax output from [0.9902, 0.0098] to [0.000, 1.000]. (Bottom): minimal perturbation to flip Random Forest's classification from '3' to '8', and surrogate softmax output from [0.000,1.000] to [0.8910, 0.1090]. For the central images, blue indicates pixels which are substantially reduced in value, red pixels substantially increased in value. (Substantially means an increase or decrease of > 0.2, where MNIST pixel values have been scaled to lie in [0,1]).

negation is necessary. If presenting FIMAP's outputs as the perturbations required to cause a *misclassification*, the outputs of the perturbation network can be directly reported.

In this simple continuous space, the explanations output by FIMAP correspond closely with those output by LIME. The mean cosine similarity between the 2000 LIME explanations and the 2000 FIMAP explanations is 0.936. FIMAP has two clear advantages over LIME on this sort of data; it is faster, and it indicates how close an instance is to a decision boundary, which can be treated as a proxy to how confident we should be in the black box classifier's prediction. In terms of speed, the time for LIME to generate the 2000 explanations above was 214 seconds. FIMAP took 53.5 seconds to train once, and subsequently generated 2000 explanations in 1.32e-2 seconds.

As a consequence of regularising to return the *minimal* perturbation, instances which have perturbations with small magnitude relative to the average for the dataset, are instances close to a decision boundary. This might be an indication that these instances are worth further examination; either by a preferred but slower explanation model, or directly by a user attempting to diagnose the behaviour of the black box model. In Figure 2 (right), the smallest 10% of perturbation vectors have been highlighted in red, and clearly track the decision boundary. Removing them from the cosine comparison improves mean cosine similarity with LIME's explanations to 0.964. The appendix contains an extended investigation of FIMAP's ability to highlight regions of interest, in comparison to LIME.

Continuous Features - MNIST Pixel Perturbation

To demonstrate FIMAP's performance on more complex data with much larger feature spaces, we trained a 200 tree Random Forest classifier on a two-class subset of the MNIST dataset, where the classes were '8' and '3'. The Random Forest achieved an f-score of 0.98. We then trained FIMAP on the label provided by the Random Forest. The structure of both surrogate and perturbation networks was identical to that in the simple synthetic cases detailed above (see Data and Methods section for an overview). The surrogate model achieved a recovery accuracy of 0.983 on the test set, and the perturbation network an adversarial accuracy of 0.975 on the test set.

Figure 3 shows examples of the minimal perturbation required to change the surrogate's classification to the incorrect label. Both instances shown also flip the classification of the unseen Random Forest. As can be seen, FIMAP has learned to either remove part of the left hand strokes of 8s, or partly fill the gaps for 3s. That it does not do so fully is due to its remit to recover minimal perturbations - it does not need to fully remove or redraw the relevant part of the letter to flip the classifier's decision.

Categorical Features

Lastly, we show how FIMAP handles a mixture of continuous and categorical variables. On a subset of the UCI

	Age	Education Years	Weekly Hours	Work Class Type	Marital Status	Occupation	Race	Sex	Model Class
Example 1: True Features	36.0	9.0	40.0	Private	Married	Craft/ Repair	Black	Male	<=$50K
Perturbation	40.5	9.0	42.7	Private	Married	Craft/ Repair	White	Male	>$50K
Example 2: True Features	32.0	9.0	40.0	Private	Married	Sales	White	Female	<=$50K
Perturbation	39.8	5.0	42.0	Private	Married	Sales	White	Male	>$50K
Example 3: True Features	31.0	11.0	40.0	Private	Married	Exec/ Managerial	White	Male	>$50K
Perturbation	39.7	5.3	41.8	Private	Married	Sales	White	Male	<=$50K

Table 1: Example instances (True Features) and their perturbations generated by FIMAP on a subset of the UCI Adult dataset. Model Class indicates the classification assigned by the underlying Random Forest classifier.

Adult Dataset (Dua and Graff 2017), our Random Forest achieved an f-score of 0.81, and FIMAP a surrogate accuracy of 0.882, and an adversarial accuracy of 0.853.

Table 1 shows three example perturbations produced by FIMAP. In each, we penalised flipping more than one categorical variable at a time, to encourage the minimal perturbations to highlight particular categories that contribute most substantially to the classification (in the UCI Adult dataset, classification is between individuals predicted to earn >$50K and those predicted to earn <=$50K).

Example 1 in Table 1 shows an instance where FIMAP deduced that changing the race of the individual from 'Black' to 'White' was sufficient to flip the classification. Example 2 shows an instance where changing sex from 'Female' to 'Male' similarly increased predicted earning potential. Example 3 shows an instance where changing an individual's occupation from 'Managerial' to 'Sales' is sufficient to push them below the boundary. Note that for Examples 2 and 3, Education years are also changed substantially. We hypothesise that this is due to correlations in the decision boundary of the classifier between variables; further work would be required to pull apart the exact mechanism at work here.

An open question when dealing with data with a mixture of variables is: to what extent are perturbations comparable? In Example 1 of Table 1, FIMAP increases the age of the individual by around 4 years, and, as discussed, changes their race from 'Black' to 'White'. Which is a more substantial change? When searching for a *minimal* perturbation, the model's relative weighting of age (a continuous variable) and marital status (a discrete variable) is dependant on the value ascribed to the relative magnitudes of λ and η, the hyperparameters weighting the minimsing regularisation terms for continuous and discrete variables, respectively (see Equation (5)).

In practice, we found that it was necessary to set η to around an order of magnitude larger than λ (the values for the above perturbations were $\eta = 2.0$, $\lambda = 0.1$), to prevent the model from making such substantial changes to the categorical variables of each instance so as to be uninformative. With this setting, we found that changes to marital status and occupation dominated the minimal perturbations for those individuals who were already close to the boundary. Both Example 1 and Example 2 in Table 1, for instance, remain white and male. However, Example 3 requires substantial changes to almost every variable to convince the classifier to change its decision.

Conclusion

Whilst we have compared ourselves to the literature on model-agnostic instance-wise explanation, we are not necessarily in competition with it. FIMAP can be thought of an additional tool in the model development toolbox. It possesses a substantial speed advantage over comparative sampling-based methods, and it provides novel functionality in its ability to indicate regions of data space where further investigation of the behaviour of the underlying classifier is warranted.

FIMAP can be thought of as a novel approach in that it proposes *minimal adversarial perturbations* as a useful explanatory tool. This also aligns it with the literature on counterfactuals as explanations (Wachter, Mittelstadt, and Russell 2017), as the minimal adversarial perturbation can also be thought of as the minimal *counterfactual direction* - the direction in which one could perturb an instance to cause a classifier to change its classification.

Finally, as we have shown in continuous feature spaces, FIMAP also produces results comparable (under a change of sign) to the output of additive feature attribution methods, such as LIME and SHAP. Though FIMAP is not itself an additive attribution method, we can think of FIMAP as demonstrating the relationship between two distinct paradigms of explanation; that the vector of feature contributions to the output of some model f for some instance $\boldsymbol{x}^{(n)}$ is the negative of the direction of perturbation to that instance required to recover its nearest counterfactual $\boldsymbol{p}^{(n)}$.

Acknowledgements

Initial work was done by MCR and MAS whilst on internship at QuantumBlack in 2019, and subsequent fellowships. MAS was further funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - Project-ID 414984028 - SFB 1404. MCR was supported in part by the EPSRC Centre for Doctoral Training in Data Science,

funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1) and the University of Edinburgh.

References

Bhatt, U.; Weller, A.; and Moura, J. M. 2020. Evaluating and Aggregating Feature-based Model Explanations. *arXiv preprint arXiv:2005.00631* .

Bhatt, U.; Xiang, A.; Sharma, S.; Weller, A.; Taly, A.; Jia, Y.; Ghosh, J.; Puri, R.; Moura, J. M.; and Eckersley, P. 2019. Explainable Machine Learning in Deployment. *arXiv preprint arXiv:1909.06342* .

Chen, J.; Song, L.; Wainwright, M. J.; and Jordan, M. I. 2018. Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. *arXiv preprint arXiv:1802.07814* .

Cheng, M.; Le, T.; Chen, P.-Y.; Yi, J.; Zhang, H.; and Hsieh, C.-J. 2018. Query-efficient hard-label black-box attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457* .

Dabkowski, P.; and Gal, Y. 2017. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems*, 6967–6976.

Dhurandhar, A.; Chen, P.-Y.; Luss, R.; Tu, C.-C.; Ting, P.; Shanmugam, K.; and Das, P. 2018. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in neural information processing systems*, 592–603.

Dombrowski, A.-K.; Alber, M.; Anders, C. J.; Ackermann, M.; Müller, K.-R.; and Kessel, P. 2019. Explanations can be manipulated and geometry is to blame. *arXiv preprint arXiv:1906.07983* .

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository. URL http://archive.ics.uci.edu/ml.

Etmann, C.; Lunz, S.; Maass, P.; and Schoenlieb, C. 2019. On the Connection Between Adversarial Robustness and Saliency Map Interpretability. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 1823–1832. Long Beach, California, USA: PMLR.

Fong, R. C.; and Vedaldi, A. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, 3429–3437.

Ghorbani, A.; Abid, A.; and Zou, J. 2019. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3681–3688.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* .

Karpathy, A.; Johnson, J.; and Fei-Fei, L. 2015. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078* .

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.; et al. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11): 2278–2324.

Lipton, Z. C. 2016. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490* .

Lundberg, S. M.; Erion, G. G.; and Lee, S.-I. 2018. Consistent Individualized Feature Attribution for Tree Ensembles. *arXiv preprint arXiv:1802.03888* .

Lundberg, S. M.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, 4765–4774.

Molnar, C. 2019. *Interpretable Machine Learning*. bookdown.

Montavon, G.; Samek, W.; and Müller, K.-R. 2018. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing* 73: 1–15.

Olah, C.; Mordvintsev, A.; and Schubert, L. 2017. Feature visualization. *Distill* 2(11): e7.

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

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. ACM.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2018. Anchors: High-Precision Model-Agnostic Explanations. In *AAAI Conference on Artificial Intelligence*.

Sharma, S.; Henderson, J.; and Ghosh, J. 2019. CERTIFAI: Counterfactual Explanations for Robustness, Transparency, Interpretability, and Fairness of Artificial Intelligence models. *arXiv preprint arXiv:1905.07857* .

Shrikumar, A.; Greenside, P.; and Kundaje, A. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 3145–3153. JMLR. org.

Singla, S.; Wallace, E.; Feng, S.; and Feizi, S. 2019. Understanding Impacts of High-Order Loss Approximations and Features in Deep Learning Interpretation. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 5848–5856. Long Beach, California, USA: PMLR.

Su, J.; Vargas, D. V.; and Sakurai, K. 2019. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation* 23(5): 828–841.

Wachter, S.; Mittelstadt, B.; and Russell, C. 2017. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GPDR. *Harv. JL & Tech.* 31: 841.

White, A.; and Garcez, A. d. 2019. Measurable Counterfactual Local Explanations for Any Classifier. *arXiv preprint arXiv:1908.03020* .

Yeh, C.-K.; Hsieh, C.-Y.; Suggala, A. S.; Inouye, D.; and Ravikumar, P. 2019. How Sensitive are Sensitivity-Based Explanations? *arXiv preprint arXiv:1901.09392* .

Zhao, Z.; Dua, D.; and Singh, S. 2017. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342* .