Sensible local interpretations via class-weight uncertainty and conditional perturbation

Chandan Singh
University of California, Berkeley

David Ruhe
SURFsara

Giovanni Cinà
Pacmed BV

Michele Tonutti*
Pacmed BV

Abstract

Local interpretations are critically important to make safe and informed decisions in many real-world applications of machine learning, such as medicine. We identify three desirable types of local interpretations for classifiers: uncertainty, sensitivity, and contribution, each of which provides crucial information. We develop model-agnostic methods to obtain these interpretations in a way which is easy to understand, even for users without deep knowledge of machine learning (e.g. medical practitioners). To calculate uncertainty, we take the difference between an over-confident and under-confident classifier, trained by weighting different types of errors in their loss functions. Sensitivity and contribution are both derived from changing the value of a feature, keeping all other features fixed. Combining these methods yields uncertainty for the feature-importance scores themselves. Various simulations and experiments on many datasets (including the MIMIC critical care dataset) demonstrate that the proposed methods are effective and yield insights into individual predictions.[1]

1 Introduction

Modern machine learning models have demonstrated strong predictive performance across a wide variety of settings. However, these models have become increasingly difficult to interpret, limiting their use in fields such as medicine [1]. Moreover, the use of such models has come under increasing scrutiny as they struggle with issues such as fairness [2] and regulatory pressure [3]. To address these concerns, research in interpretable machine learning and uncertainty has received an increasing amount of attention [4, 5].

With the increasing complexity of modern interpretation techniques, it is often unclear which interpretation approach is best-suited to a particular audience and task. Moreover, new interpretation techniques often exhibit undesired behavior, drawing much criticism [6, 7]. In this work, we propose sensible, effective metrics understandable to a non-technical user (e.g. a medical practitioner) for three types of local interpretation: (1) uncertainty, (2) sensitivity, and (3) contribution (see Fig 1). Each conveys unique, important information. Uncertainty shows how much one can trust the model's prediction; sensitivity yields the effect of changing a feature; and contribution conveys the relative importance of a feature. Moreover, combining these methods, one can obtain uncertainty estimates not just for the prediction, but for the contribution and sensitivity scores as well. This allows both a medical practitioner or a model developer to know when the interpretations can be trusted.

This work develops three main novel contributions, which are all connected. The first is class-weight uncertainty, a simple method for obtaining uncertainty via weighting classes in a loss function. The second is a way to obtain intuitive feature importances for sensitivity and contribution based on ICE curves, and the third is uncertainty for feature importances via class-weighting. As a secondary con-

*Authors listed in order of contribution.

[1]Code with a simple API for reproducing all results will be made available on github.

Preprint. Under review.

tribution, we also provide algorithms and a software package to calculate and (interactively) visualize interpretations of this form. Together, these methods yield a comprehensive local interpretation which is understandable to a non-technical audience, such as medical practitioners.

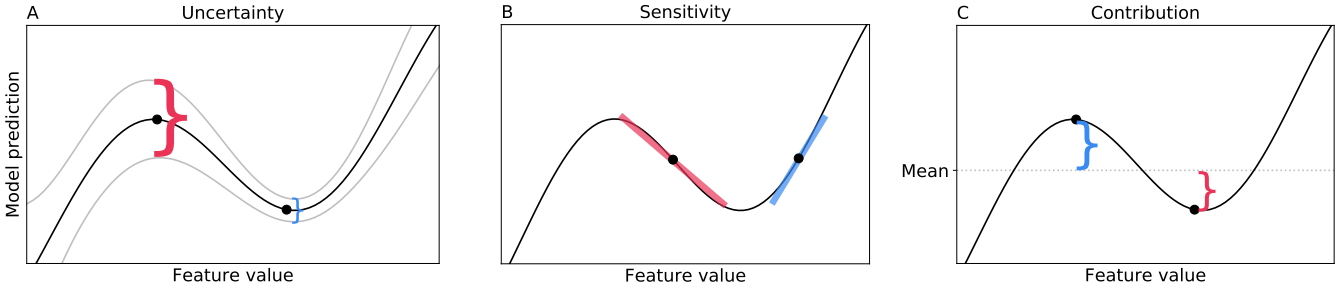

Figure 1: Prediction-level interpretation methods. (A) Uncertainty is given by a prediction interval around a point. (B) Sensitivity measures how the prediction changes in response to small perturbations of a feature. (C) Contribution measures how the model's prediction differs from the mean prediction when all but one feature is fixed.

The work here is evaluated via a series of simulations and experiments on a variety of real-world datasets. Most emphasis is given to tabular data, given its relevance to healthcare. The proposed uncertainty method is specific to classification, while sensitivity and contribution can readily be applied to regression as well. This work is completely model-agnostic and can be directly applied to any model or dataset.

2 Background

This section briefly overviews related work organized into the three approaches to provide local interpretations shown in Fig 1, particularly with respect to healthcare applications.

Uncertainty There is a large literature on using Bayesian methods to ascertain uncertainty estimates [8–10]. However, these methods generally require significant modification to the modelling process, often making training more difficult and potentially sacrificing some predictive accuracy.

More similar to the work here are methods which do not fundamentally change the classifier (e.g. by using ensembling to ascertain uncertainty [11]). Additionally, one can obtain uncertainty directly from a model's prediction by calibrating classifiers after training (e.g. Platt scaling [12], which fits a logistic regression model to a classifier's scores).

Finally, the work here extends the intuition behind quantile regression [13], where regressing on different quantiles can provide prediction intervals for a regressor, to work in classification settings as well.

Sensitivity Methods for calculating sensitivity have long been important for understanding feature importance [14]. These techniques consist of different methods for approximating a local derivative [15, 16]. Notably, LIME [17] is a very popular recent method which effectively fits a locally linear model approximating the sensitivity of a feature for tabular data. Numerous works have extended the intuition behind sensitivity to interpret neural networks based on their local gradients [18–21]. However, recent work has noticed that such feature importance scores are fragile [22, 6]; to our knowledge, not much attention has been dedicated to quantifying and understanding this fragility [23, 24].

Contribution Contribution scores take many forms but all provide some information about how much the prediction changes after conditioning on the value of a feature [25–27]. At the single-prediction level, different types of contribution scores have been developed to compute contribution for neural networks, both with interactions [28, 29] and without [30, 31]. Notably, the popular SHAP score [32] measures the change in the mean prediction after conditioning on all possible subsets of features.

It should be noted that these methods are different than global (dataset-level) methods such as the permutation importance [26], which provide a single scalar importance over the entire dataset. Additionally, many methods exist to calculate importance curves, such as partial dependence plots (PDP) [33, 34], accumulated local effects plots [35], and disentangled attribution curves [36].

3 Methods

This section details the three methodological contributions of this work: class-weight uncertainty (Sec 3.1), sensitivity/contribution scores (Sec 3.2), and uncertainties for feature importances (Sec 3.3).

3.1 Class-weight uncertainty

We desire an uncertainty score which can be easily obtained and is intuitive. To do so, we manipulate only the loss function of a model to obtain an underconfident and overconfident model which then provide prediction intervals to be used as uncertainty. Specifically, we assign a weight w_c to each class c in the loss function:

$$\mathcal{L}'(X, y) = \sum_{c \in C} w_c \mathcal{L}(X_c, Y_c) \tag{1}$$

where (X_c, Y_c) are the sets of points for which $Y_c = c$.

Refitting the model with different weights w_c yields multiple models. For binary classification, some of these models are "overconfident" (i.e. have a high recall but low precision) and some are "underconfident" (i.e. have a high precision but low recall). From these models, we select a best model, and an overconfident and underconfident model to yield intervals for the best model's prediction, similar to Fig 1A. For a given point x, the uncertainty is then defined as the difference between the prediction of the overconfident model and the underconfident model:

$$\text{uncertainty}(x) = \hat{f}_{\text{overconfident}}(x) - \hat{f}_{\text{underconfident}}(x) \tag{2}$$

Note that this is similar to a Bayesian inference setting in which we sample models and use predictive variance as the uncertainty setting. In our case, we sample models and train them directly to capture this interval by making them artificially represent the over- and underconfident sampled models in the Bayesian case. In the case of cross-entropy loss the underconfident model is still incentivized to make confident predictions, and vice versa. This ensures that we are not just scaling the predictions globally, but the models locally assess which datapoints can be confidently predicted and which cannot.

Choice of class-weights The choice of the weights is application-specific and can be chosen so that the over-/underconfident models achieve some user-specified precision-recall values (example given in Sec 4.2). In this way, one knows precisely how precision and recall trade off within the provided prediction interval. A simple suggestion to start is to merely double the weight of the positive/negative class. Note that this formulation is specific to classification; for regression, one can penalize overconfident/underconfident predictions by weighting positive and negative residuals differently, rather than weighting classes.

Restricted refitting For some complex models, such as deep neural networks or boosted trees, the refitting procedure can be computationally expensive. To avoid these issues, rather than retraining from scratch, we initially train one model without class-weighting, then initialize all other models from the resulting model. In this way, refitting is extremely fast and can be achieved with only a few gradient steps. Moreover, for additional speed-ups, many layers of a neural network (e.g. all but the last) can be frozen after training the initial model. In other words, all feature extraction procedures can be left as is, and only the classifier should be trained with a weighted loss. This can help to mitigate recent work which observed that some neural networks are insensitive to importance weighting [37].

Approximation via bootstrap sampling Some models, such as boosted trees, are able to achieve exactly zero training loss and their decision boundaries may not change as a result of the class re-weighting. For such models, we propose to re-fit the model via bootstrap samples according to the class weights.

Extension to multi-class In multi-class classification, the above procedure would require calculating an overconfident and underconfident model for each class, which quickly becomes computationally expensive. To scale to problems with many classes, a simpler approach involves training only one overconfident classifier, which penalizes/rewards confident predictions (similar to [38]):

$$\mathcal{L}'(X, y) = \mathcal{L}(X, y) + \alpha \, KL\left(\mathcal{U}(y)||\hat{f}(X)\right) \tag{3}$$

Here, $\mathcal{U}(y)$ is the uniform distribution, KL refers to the KL-divergence, and α can be either positive or negative, unlike in previous work.

3.2 ICE-contribution and ICE-sensitivity

ICE-contribution and ICE-sensitivity both are calculated from the individual conditional expectation (ICE) curve [34], which stems from partial dependence plots [33]. The ICE curve is calculated by fixing the values of all features except for one feature X_i. Formally, the ICE curve function given model $\hat{f}$, set of features not including X_i: $C = \{X_j \in X | X_j \neq X_i\}$, and point of interest $\mathbf{x}^*$ is defined as:

$$\hat{f}_{i,\mathbf{x}^*}(x) = \hat{f}(X_i = x, X_C = \mathbf{x}^*_\mathbf{C}) \tag{4}$$

The ICE curve shows useful information about the sensitivity of the model to this feature. However, varying X_i can produce many points which do not fit the data distribution, resulting in misleading interpretations. To avoid this issue, we obtain a conditional distribution for the feature to be explained by conditioning on the values of the other features: $P(X_i|X_C = \mathbf{x}^*{}_C)$. If the features are assumed to be independent, this can be estimated very easily using standard density estimation techniques. For more complex domains, a model may be used to calculate this distribution (e.g. a model which predicts a missing word given the rest of the words, as is done in modern language models such as BERT [39]). These sampling densities can be displayed on the curve, and are used in the calculation of the ICE-Contribution score.

ICE-Contribution score The ICE-contribution score for feature X_i is defined as the expected difference from the mean ICE value, over the conditional distribution for X_i:

$$\text{ICE-contrib}_i(\mathbf{x}^*) = \hat{f}_{i,\mathbf{x}^*}(x) - \mathbb{E}_{X_i|X_C=x^*_C}[\hat{f}_{i,\mathbf{x}^*}(x)] \tag{5}$$

This has the intuitive interpretation as the change in the prediction due to the presence of this feature, holding all other features constant. As a result, it is useful for determining the relative influence of features and an effective first step in an interpretability pipeline may be to rank the features based on this score.

The ICE-contribution score is closely related to other types of contribution scores, such as SHAP [32] and permutation importance [26]. ICE-contribution deals with interactions by conditioning on all features except the one of interest, making it easily understandable. In contrast, SHAP and other complex scores include all the interactions between features, obscuring whether single features contribute individually or through interactions. In addition, the use of conditional expectation in (5) prevents the method from suffering many of the usual dangers of post-hoc explanation, where the function is evaluated on data examples which do not follow the original data distribution [40, 41] (more details in Sec S2).

ICE-Sensitivity score The second score measures the sensitivity of the model's prediction to small changes in feature value (holding all other features fixed). For a linear regression model, this would be precisely the coefficient for the feature X_i. To compute this for continuous variables, we simply take the partial derivative, which can easily be approximated numerically. For categorical variables,

we use one-hot encoding to make any feature binary and then simply report the change with respect to changing the value of each binary variable.

Care must be taken with models which represent step functions, such as tree-based models, where the local derivative is often zero. For these models, we report the sensitivity by approximating the local derivative extending to the region where the prediction changes. We find the smallest change Δ_{X_i} for which the model prediction changes (which can be easily found by looking at the splits of the tree). Then we can approximate the local derivative numerically: $f(x + \Delta_{X_i}) \,/\, \Delta_{X_i}$. We further refine this approximation by taking the mean of this derivative in the positive and negative direction (calculated by restricting Δ_{X_i} to be positive and negative, respectively).

3.3 Uncertainty for importance scores

Uncertainty for importance scores can be obtained by calculating the standard deviation of the importance between the scores for the different models obtained by class-weighting. Intuitively, importance scores with more variation are more uncertain. This can be done for both the ICE-Contribution and ICE-Sensitivity score, and in fact can be applied more generally to any importance score, such as SHAP or LIME.

4 Results

4.1 Uncertainty on real data

To evaluate the efficacy of class-weight uncertainty, we measure its ability to identify when the model makes errors. We calculate a curve which represents the test AUC for the classifier which is repeatedly evaluated by adding points to the testing dataset, sorted by their uncertainty (least uncertain points are added first). Table 1 reports the AUC of each of these curves. In all cases, scikit-learn defaults are used: for a multi-layer perceptron (MLP), this entails two layers, hidden layer size of 100, ReLu activation function, and ADAM optimizer. For gradient boosting, 100 estimators and deviance loss are used. All results in this section are on the test set (which consists of a random 25% of the total dataset).

Table 1 shows that class-weight uncertainty effectively captures uncertainty. For each model, the class-weight column should be compared to the baseline column. The baseline uncertainty measure is taken to be asymmetric entropy, as it is an intuitive baseline which is model-agnostic and works in class-imbalanced problems [42]. The class-weight metrics are almost always higher than the baseline, sometimes by more than eight percent.

Table 1: Ability of uncertainty to identify when a model makes errors (higher is better). Anything which provides at least an 8 percent boost is shown in bold.

	Logistic Regression		2-layer MLP		Gradient Boosting	
Dataset	Class-weight	Baseline	Class-weight	Baseline	Class-weight	Baseline
hill valley with noise	74.82	71.47	**64.38**	54.16	45.42	45.23
agaricus lepiota	94.42	94.41	96.76	96.76	**96.76**	67.70
churn	**61.62**	53.38	**76.01**	65.20	86.62	81.82
clean2	**96.00**	76.18	**93.58**	36.00	15.88	17.15
coil2000	57.36	57.66	55.11	52.58	**74.46**	64.32
diabetes	59.75	58.80	51.60	49.57	56.71	57.62
dis	48.96	42.87	**70.42**	37.65	**50.09**	41.88
german	51.11	48.40	**50.61**	28.45	59.98	52.50
hypothyroid	26.30	29.73	52.97	57.60	**61.79**	45.26
tokyo1	67.55	65.66	68.39	65.65	**71.87**	57.41
Mean	63.79	59.86	**67.98**	54.36	**61.96**	53.09

4.2 MIMIC Case Study

In this section, we show how the proposed methods yield useful local interpretations on the MIMIC dataset [43], which consists of data for predicting the mortality risk of patients in the intensive care

unit. We analyze a highly predictive neural-network model trained with one dense layer of hidden size 64 and ReLU activation. By varying the class-weight, we obtain three models, each with different precision-recall values. In the case of MIMIC, a high recall is more important than a high precision (as we do not want to miss a patient with high mortality risk), and the chosen models reflect this: the unweighted model has precision-recall values of 0.63 and 0.60, respectively, the overconfident model has values of 0.54 and 0.66, and the underconfident model has values of 0.70 and 0.52.[2] These models were chosen via cross-validation, tuning class-weight as a hyperparameter.

Fig 2A shows how class-weight uncertainty accurately captures when the model will fail. The loss of the model increases as we add more uncertain points to the test set. Fig 2B shows how the class-weight uncertainty varies for different prediction values. As expected, uncertainties are generally highest around predictions of 0.5. Nevertheless, class-weight uncertainty is much more expressive than the uncertainty of the prediction itself: it achieves an AUC of 0.87 when predicting false predictions, whereas the uncertainty measured in terms of asymmetric entropy only achieves 0.58 AUC.

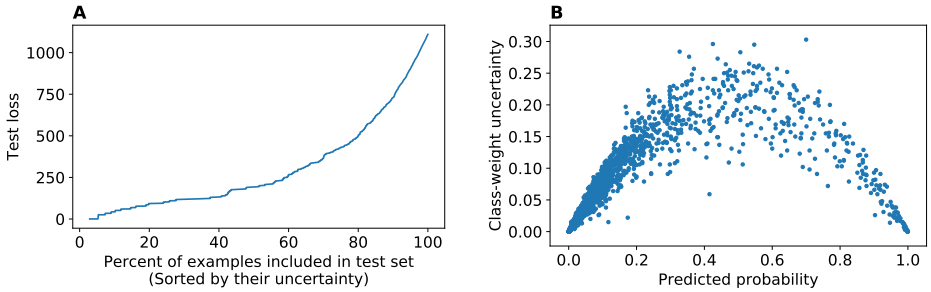

Figure 2: MIMIC results. **A**. Test loss increases as more uncertain examples are included, showing that the uncertainty effectively identifies misclassifications. **B**. Class-weight uncertainty is similar, but not quite the same as the uncertainty capture by the prediction itself.

In practice, one can use an interface, such as that in Fig 3[3], to comprehensively understand how a model made a prediction. In addition to the prediction and uncertainty given at the top of the interface, the table on the left shows the features, their values, importance scores, and the standard deviations of these scores (over the different trained models). The user can then sort by different scores and select different features to examine in more detail via ICE curves in the right panel. For this particular prediction, we can see that a few features have a much higher contribution than the rest, and one feature is much more sensitive than others.

4.3 Intuitive feature importances

The main advantage of both the ICE-contribution and ICE-sensitivity scores over other methods is their simplicity. They both yield clear information about the effect of a feature when holding all other features constant. This information can clearly be used to find, for example, when the function is fairly smooth around a point or when a feature's value has an extreme effect on the prediction. Since this is intuitive for tabular data, we focus our evaluation instead on less clear aspects of these scores.

Text example To extend the intuition of these scores beyond tabular data, we present an example in text classification. Building models directly from text is an active area of research in many domains such as medicine [44]. Here, we illustrate the simple case of sentiment classification by finetuning a pre-trained BERT model [39] on the Stanford Sentiment Treebank [45] to predict whether a movie review is positive or negative.[4] Fig 4 shows the prediction for one particular sentence: "This great movie was very ___", where the blank is filled in by different words. The words on the x-axis are

[2] All models use the same threshold, which is chosen to yield the correct amount of positive class predictions for the canonical model.

[3] See this interface interactively in the supplementary material.

[4] Pre-trained model from `https://github.com/huggingface/pytorch-transformers`.

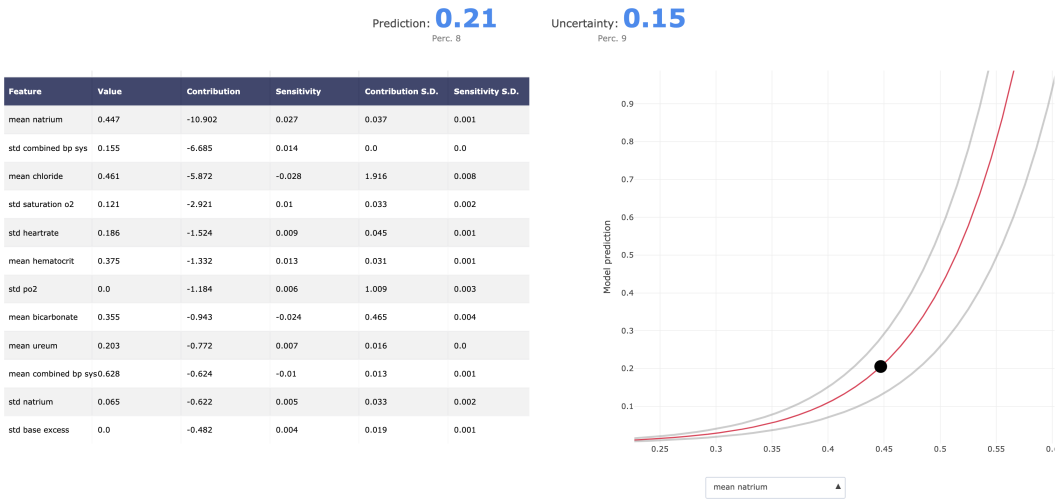

Figure 3: Screenshot of interface for interpreting a single prediction, displaying the proposed scores. Plot on the right shows the model prediction as one feature is varied, holding all other features constant.

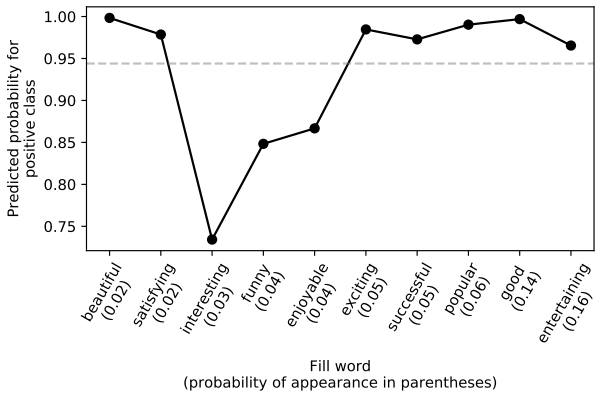

Figure 4: Figure shows positive-class predicted probability for the sentence “This great movie was very ____”.

sorted by the probability that they fill in the blank, as predicted by the BERT language model. This curve provides a natural way to visualize sensitivity for textual features which are not continuous.

Filling in the sentence with the word ‘good’ yields a relatively high prediction of 0.997. When calculating contribution, it is important to note that the mean prediction of 0.813 (when filling in the blank word randomly) is already quite high because the rest of the sentence is quite positive. Moreover, the conditional mean (0.944, dotted gray line) is even higher, as the top predicted words for filling in the blank are already fairly positive (see axis of Fig 4). As a result, adding the word ‘good’ does little to change the expected prediction of the model whereas filling in with an unexpected word, such as ‘bad’, causes a dramatic change (changes the prediction to 0.003).

Simulation study Fig 5 shows results for simulations aimed at evaluating feature importance scores. In order to provide a ground truth, these simulations are deliberately simple. All simulations have Gaussian i.i.d. features and a single response variable. The response variable is computed as the logit of a probability which is a linear combination of a few features. Each simulation varies parameters including the number of training data points, the number of features, the number of features used to generate the response, and the coefficients of these features, and the variance of the noise (full simulation details and more simulations including non-linear data generating processes in Sec S4).

For each simulation, we fit a gradient boosting (gb) model, a multi-layer perceptron model (mlp2) and a logistic regression model, again using only scikit-learn defaults.

The coefficients of the relevant features in the simulation are taken to be the ground truth feature importances of the model. Then, for each importance metric, three metrics are measured. *Fraction Correct Signs* refers to the fraction of the signs of the important features which are the same as the signs of the importance measures. *Fraction Intersect* refers to the fraction of the truly important features which were ranked highest based on the absolute value of the importance measure. Finally, *Rank Corr* refers to the Spearman's rank correlation between the importances assigned to the features and their ground truth importances. *Acc* reports the test accuracy of the model for the simulation.

In these relatively simple simulations, ICE-sensitivity generally seems to outperform the other metrics. Additionally, the ice-contrib score is very similar to the SHAP score, suggesting that both capture similar information. Finally, it is worth noting that by looking at information from different importance scores, we can do better than looking at any one score in isolation. These simulations are certainly not comprehensive, but do help to illustrate simple situations in which these scores are effective.

score	acc	ice-contrib			ice-sensitivity			lime			shap		
metric		Fraction Correct Signs	Fraction Intersect	Rank Corr	Fraction Correct Signs	Fraction Intersect	Rank Corr	Fraction Correct Signs	Fraction Intersect	Rank Corr	Fraction Correct Signs	Fraction Intersect	Rank Corr
Sim 0 gb	0.78	0	1	-1	1	0.5	1	0	1	1	0	1	-1
Sim 0 logistic	0.8	0	1	-1	1	1	1	0	1	1	0	1	-1
Sim 0 mlp2	0.79	0	1	-1	1	1	1	0	1	1	0	1	-1
Sim 1 gb	0.77	0	0.5	-1	1	1	-1	0	1	1	0.5	1	-1
Sim 1 logistic	0.79	0.5	1	-1	1	1	1	0.5	1	1	0.5	1	-1
Sim 1 mlp2	0.76	0	0.5	-1	1	1	1	0.5	1	1	0.5	1	-1
Sim 5 gb	0.83	0	1	-0.5	1	1	1	0	1	1	0	1	-0.5
Sim 5 logistic	0.86	0	1	-0.5	1	1	1	0	1	1	0	1	-0.5
Sim 5 mlp2	0.84	0	1	-0.5	1	1	1	0	1	1	0	1	-0.5
Sim 6 gb	0.89	0	1	-1	1	1	-1	0	1	1	0	1	-0.8
Sim 6 logistic	0.9	0	1	-0.8	1	1	1	0	1	1	0	1	-0.8
Sim 6 mlp2	0.9	0	1	-1	1	1	1	0	1	1	0	1	-0.8

Figure 5: Simulation results for feature importance scores. ICE-Contribution and ICE-Sensitivity seem to perform just as well (or slightly better) than popular alternatives SHAP and LIME. Blue scores are best, white are neutral, and red are worst (different columns are on different scales).

5 Discussion and future work

The work here provides simple, but surprisingly effective baselines for local interpretation. Many things can be added to this baseline to improve interpretations for specific applications. For example, a good practice would be to look at many such interpretations and perhaps their nearest neighbors as well. Additionally, while we look here mainly at individual features, one could extend this notion to look at groups of features as well.

Some concepts introduced here apply more generally. The idea of a weighted conditional expectation curve can be used in generality. This helps combat its main shortcoming: the out-of-distribution sampling issue. Since we are looking at only one ICE plot here, we can extend this to look at interactions between two variables (whereas the original ICE plot would require looking at several overlaid surfaces in order to understand interactions). While we mainly emphasized getting uncertainty scores for the ICE-contribution and ICE-sensitivity scores, one could do precisely the same thing for other scores, such as SHAP. We hope that the work here can help improve the use of local interpretations and lead to more transparent machine learning in practice.

References

[1] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen AWM van der Laak, Bram van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[2] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226. ACM, 2012.

[3] Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a" right to explanation". *arXiv preprint arXiv:1606.08813*, 2016.

[4] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*, 2019.

[5] Finale Doshi-Velez and Been Kim. A roadmap for a rigorous science of interpretability. *arXiv preprint arXiv:1702.08608*, 2017.

[6] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pages 9505–9515, 2018.

[7] Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[8] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[9] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[10] Peter Congdon. *Applied bayesian modelling*, volume 595. John Wiley & Sons, 2014.

[11] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

[12] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[13] Roger Koenker and Gilbert Bassett Jr. Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50, 1978.

[14] Donglai Wei, Bolei Zhou, Antonio Torrabla, and William Freeman. Understanding intra-class knowledge inside cnn. *arXiv preprint arXiv:1507.02379*, 2015.

[15] Dan G Cacuci and Mihaela Ionescu-Bujor. A comparative review of sensitivity and uncertainty analysis of large-scale systems—ii: statistical methods. *Nuclear science and engineering*, 147(3):204–217, 2004.

[16] S Kucherenko et al. Derivative based global sensitivity measures and their link with global sensitivity indices. *Mathematics and Computers in Simulation*, 79(10):3009–3017, 2009.

[17] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[18] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[19] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *ICML*, 2017.

[20] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *See https://arxiv. org/abs/1610.02391 v3*, 7(8), 2016.

[21] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.

[22] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. *arXiv preprint arXiv:1710.10547*, 2017.

[23] Yujia Zhang, Kuangyan Song, Yiming Sun, Sarah Tan, and Madeleine Udell. "why should you trust my explanation?" understanding uncertainty in lime explanations. *arXiv preprint arXiv:1904.12991*, 2019.

[24] Jiayun Dong and Cynthia Rudin. Variable importance clouds: A way to explore variable importance for the set of good models. *arXiv preprint arXiv:1901.03209*, 2019.

[25] Carolin Strobl, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis. Conditional variable importance for random forests. *BMC bioinformatics*, 9(1):307, 2008.

[26] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[27] Ilya M Sobol. Sensitivity analysis for non-linear mathematical models. *Mathematical modelling and computational experiment*, 1:407–414, 1993.

[28] Chandan Singh, W James Murdoch, and Bin Yu. Hierarchical interpretations for neural network predictions. *arXiv preprint arXiv:1806.05337*, 2018.

[29] W James Murdoch, Peter J Liu, and Bin Yu. Beyond word importance: Contextual decomposition to extract interactions from lstms. *arXiv preprint arXiv:1801.05453*, 2018.

[30] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

[31] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[32] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4768–4777, 2017.

[33] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[34] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1):44–65, 2015.

[35] Daniel W Apley. Visualizing the effects of predictor variables in black box supervised learning models. *arXiv preprint arXiv:1612.08468*, 2016.

[36] Summer Devlin, Chandan Singh, W James Murdoch, and Bin Yu. Disentangled attribution curves for interpreting random forests and boosted trees. *arXiv preprint arXiv:1905.07631*, 2019.

[37] Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In *International Conference on Machine Learning*, pages 872–881, 2019.

[38] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017.

[39] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[40] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. The dangers of post-hoc interpretability: Unjustified counterfactual explanations. *arXiv preprint arXiv:1907.09294*, 2019.

[41] Giles Hooker and Lucas Mentch. Please stop permuting features: An explanation and alternatives. *arXiv preprint arXiv:1905.03151*, 2019.

[42] Djamel A Zighed, Gilbert Ritschard, and Simon Marcellin. Asymmetric and sample size sensitive entropy measures for supervised learning. In *Advances in intelligent information systems*, pages 27–42. Springer, 2010.

[43] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

[44] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746*, 2019.

[45] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

Supplement

S1 Non-linear importances in practice

In practice, there are many decisions to make regarding how to compute and use importance scores. For example, feature importance score are generally used for individual features but in many applications individual features are not interpretable. For example, consider the very common case of categorical variables, which are one-hot encoded. We often would like an importance score for the categorical variable as a whole, but ICE-contrib (and similarly, other scores such as SHAP) fragment the score between each category. Another very common occurrence is features which are constructed from a time-series, which may vary in length. For example, given a time-series of measurements of a feature such as blood pressure (bp), we may use a recurrent neural network to extract features or compute summary statistics such as $mean(bp), min(bp), max(bp), change(bp), bp^2$ over some time interval. In the end, we may want to assign importance holistically to the feature bp, or more generally in some interpretable feature space.

The naive solution to this problem is to simply sum the attribution of the individual features in the group we are interested in. In fact, for some scores, such as ICE-sensitivity, this will work, as these scores linearly combine for different groups. For ICE-contribution this is often a good approximation, but if the features interact strongly, it is better to instead treat this group of features jointly as a single feature. The formulation of ICE-contribution (and in fact, many other methods such as SHAP) then yields a single score for the group while maintaining all the properties which the method fulfills.

S2 Problems with sampling

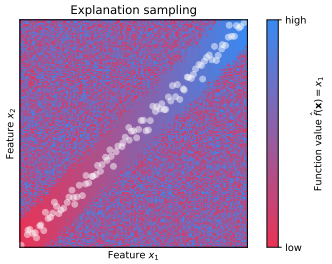

Figure S1: Out-of-distribution sampling. Sampling issues for a simple 2D function, where $\hat{f}(\mathbf{x}) = X_1$, but $X_1 = X_2$. If we don't use a conditional distribution for one feature, we calculate over regions where the function is very noisy, yielding misleading interpretations. On the other hand, If we do use a conditional sampling procedure (e.g $P(X_2|X_1)$), we may mistakenly attribute importance to X_2 even though the function depends only on X_1. The solution proposed here is to (1) use ICE-Contribution to solve the first problem regarding sampling noisy points and (2) further use ICE-sensitivity to resolve issues with spurious attribution. Essentially all existing methods suffers from one of these two issues.

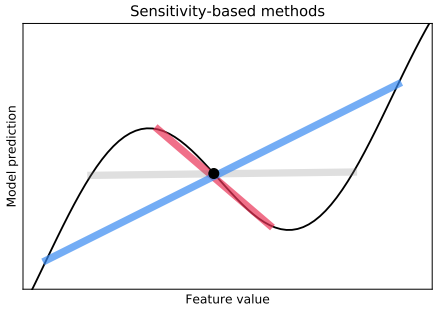

Figure S2: Sampling in the wrong window. Sensitivity-based methods are very sensitive the how big a region they are fit to.

S3 Interface examples

S4 Simulation details

Table S1: Parameters for simulations in Fig S3 and Fig 5. Feature matrix $\mathbf{X}$ is drawn IID from a standard normal distribution.

Simulation number	Number of training points	Number of features	Variance of noise	Generating function
0	7500	5	0.01	$y = 1X_0 + 2X_1[1.\ 2.\ 0.\ 0.\ 0.]$
1	7500	10	0.1	$y = 1X_0 + 2X_1[1.\ 2.\ 0.\ 0.\ 0.\ 0.\ 0.\ 0.\ 0.\ 0.]$
2	7500	5	0.001	$y = 1X_0 + 2X_1[1.\ 2.\ 0.\ 0.\ 0.]$
3	7500	10	0.001	$y = 1X_0 + 2X_1[1.\ 2.\ 0.\ 0.\ 0.\ 0.\ 0.\ 0.\ 0.\ 0.]$
4	7500	5	0.01	$y = 1X_0 + 2X_1[1.\ 2.\ 0.\ 0.\ 0.]$
5	7500	5	0.01	$y = 1X_0 + 2X_1[1.\ 2.\ 3.\ 0.\ 0.]$
6	7500	5	0.01	$y = 1X_0 + 2X_1[1.\ 2.\ 3.\ 4.\ 0.]$
7	37500	5	0	$y = 1X_0 + 2X_1[1.\ 2.\ 0.\ 0.\ 0.]$

score	acc	ice-contrib			ice-sensitivity			lime			shap		
metric		Fraction Correct Signs	Fraction Intersect	Rank Corr	Fraction Correct Signs	Fraction Intersect	Rank Corr	Fraction Correct Signs	Fraction Intersect	Rank Corr	Fraction Correct Signs	Fraction Intersect	Rank Corr
Sim 0 gb	0.78	0	1	-1	1	0.5	1	0	1	1	0	1	-1
Sim 0 logistic	0.8	0	1	-1	1	1	1	0	1	1	0	1	-1
Sim 0 mlp2	0.79	0	1	-1	1	1	1	0	1	1	0	1	-1
Sim 1 gb	0.77	0	0.5	-1	1	1	-1	0	1	1	0.5	1	-1
Sim 1 logistic	0.79	0.5	1	-1	1	1	1	0.5	1	1	0.5	1	-1
Sim 1 mlp2	0.76	0	0.5	-1	1	1	1	0.5	1	1	0.5	1	-1
Sim 2 gb	0.77	1	1	1	1	1	1	1	1	-1	1	1	1
Sim 2 logistic	0.8	1	1	1	1	1	1	1	1	-1	1	1	1
Sim 2 mlp2	0.77	1	1	1	1	1	1	1	1	-1	1	1	1
Sim 3 gb	0.75	0	1	1	1	1	1	0	1	1	0	1	1
Sim 3 logistic	0.78	0	1	1	1	1	1	0	1	1	0	1	1
Sim 3 mlp2	0.75	0	1	1	1	1	-1	0	1	1	0	1	1
Sim 4 gb	0.77	1	1	1	1	1	1	1	1	-1	1	1	1
Sim 4 logistic	0.8	1	1	1	1	1	1	1	1	-1	1	1	1
Sim 4 mlp2	0.77	1	1	1	1	1	1	1	1	-1	1	1	1
Sim 5 gb	0.83	0	1	-0.5	1	1	1	0	1	1	0	1	-0.5
Sim 5 logistic	0.86	0	1	-0.5	1	1	1	0	1	1	0	1	-0.5
Sim 5 mlp2	0.84	0	1	-0.5	1	1	1	0	1	1	0	1	-0.5
Sim 6 gb	0.89	0	1	-1	1	1	-1	0	1	1	0	1	-0.8
Sim 6 logistic	0.9	0	1	-0.8	1	1	1	0	1	1	0	1	-0.8
Sim 6 mlp2	0.9	0	1	-1	1	1	1	0	1	1	0	1	-0.8
Sim 7 gb	0.85	0	1	1	0.5	1	-1	1	1	-1	0	1	1
Sim 7 logistic	0.62	0.5	1	1	1	1	-1	0.5	1	1	0.5	1	1
Sim 7 mlp2	0.86	0	1	1	0.5	1	-1	1	1	-1	0	1	1

Figure S3: Simulation results for feature importance scores Blue scores are best, white are neutral, and red are worst (different columns are on different scales). In these simulations, ICE-Contribution and ICE-Sensitivity seem to perform just as well (or slightly better) than popular alternatives SHAP and LIME.