9th International Workshop on Climate Informatics
October 02-04, 2019
Hosted by École Normale Supérieure, Paris, France

USING CAUSAL INFERENCE TO GLOBALLY UNDERSTAND BLACK BOX PREDICTORS BEYOND SALIENCY MAPS

Christian Reimers[1,2], Jakob Runge[2], Joachim Denzler[1,3]

***Abstract*—State-of-the-art machine learning methods, especially deep neural networks, have reached impressive results in many prediction and classification tasks. Rising complexity and automatic feature selection make the resulting learned models hard to interpret and turns them into black boxes. Advances into feature visualization have mitigated this problem but some shortcomings still exist. For example, methods only work locally, meaning they only explain the behavior for single inputs, and they only identify important parts of the input. In this work, we propose a method that is also able to decide whether a feature calculated from the input to an estimator is globally useful. Since the question about explanatory power is a causal one, we frame this approach with causal inference methods.**

I. INTRODUCTION

State-of-the-art machine learning methods, especially deep neural networks, have reached impressive results in many prediction and classification tasks in computer vision and beyond. The benefit of these algorithms for many tasks in Earth system science have been discussed in [1]. One of the main challenges that arises when applying these methods is interpretability. Improved prediction performance comes at the cost of high complexity. Together with automatic feature selection introduced by deep learning makes the resulting estimators difficult to interpret largely rendering them black boxes.

However, for many applications it is important to identify the features used in a prediction or classification task. This is especially true for applications that are safety and security relevant, for example in medicine or autonomous transportation, and for applications in which predictions can not be easily verified and, therefore, we need domain experts to determine whether a predicted values makes sense, for example, climate science.

We will discuss some works that aim to make deep learning more explainable in section II. Most of those methods have at least one of the following drawbacks. First, they only give a local explanation of the estimator, meaning an explanation that is true for a specific input and inputs very close to it. In contrast to that, we aim to produce explanations, that are not only true for single inputs but global explanations, that are able to explain most inputs.

Second, they can only assign saliency to parts of the input. They can not be used to identify whether features, for example the variance of the input, that are aggregate functions of the input are used. In most real-world problems, not the inputs directly are important, but aggregate functions. If our input is, for example, a grid of sea surface temperatures in many different positions, we do not expect a single value to be important towards a prediction task but an aggregate function such as the mean or the variance of the whole grid.

Third, they do not handle confounding. They can identify features of the input that are correlated to the output of the classifier but they do not check for a causal link between the input feature and the output. Imagine a classifier that distinguishes two classes, one consisting of red circles and one consisting of green squares. If the classifier only recognizes the shape of the object, there will still be a high correlation between the color of the object and the output of the classifier. Our method respects the label of the input as a confounder.

To understand an estimator, we need to be able to reason about it. These are questions of causality studied in the field of causal inference [2] that has many applications in Earth system science [3]. Methods from this field allow us to understand which features are causing estimations on a global scale. These features do

Corresponding author: C. Reimers, christian.reimers@uni-jena.de
[1]Computer Vision Group, Friedrich Schiller University Jena, Jena
[2]Climate Informatics Group, Institute for Data Science, German Aerospace Center, Jena [3] Michael Stifel Center Jena for data-driven and simulation science, Jena

not have to be part of the input. Our method does not require any information on the estimator but treats it as a black box estimator. Hence, it can also be used in a black box setting, if the estimator is non-differentiable or if the inner workings of the estimator are completely unknown.

After we introduce some existing solutions for the problem of identifying relevant features for an automated estimator. Afterwards, we introduce the basic concepts and notations from both, machine learning and causal inference that we are using throughout this work. In the fourth section, we describe how a machine learning approach can be phrased as an SCM and explain how we can use this to identify features that are causing the estimation of the function resulting from the machine learning approach. In the fifth section, we demonstrate our approach in a toy example. Finally we discuss open questions and problems of our approach in the final chapter.

II. Related Work

In this section we introduce existing methods that are used to explain which features are used by automatic estimators.

In feature visualization, the goal is to create an input image that evokes a maximal response from a specific neuron. If we select this neuron to be in the output layer, we can find the input that is classified maximally as a specific class. This input can be understood as a prototype of the class. This technique was demonstrated in [4], [5] . Feature visualization, however, only visualizes the maximum of the function represented by the Estimator. Since the function can be non-concave and multiple maxima might exist, feature visualization is a local explanation around this maximum, while the method presented in this work is a global method explaining all estimations. Further, to apply feature visualization, information on the gradient of the estimator is needed. In contrast, our method can be applied in the black box setting where only input-output pairs are known for the estimator.

The goal of saliency maps is to highlight parts of the input that have a high influence on the output. There are multiple ways to derive the saliency of an input. The first and most straightforward approach is to use the gradient of the loss function depending on every input value or some approximation of this gradient. See [6], [7], [8] for examples of this approach. A slightly more advanced approach is to Taylor-approximate the learned function. A first order Taylor-approximation is demonstrated in [9], [10]. Higher order Taylor-approximations are used in [11], [9]. A third way to generate saliency that can also be used for non-differentiable estimators is substituting parts of the input with a neutral alternative and record the change in the output as demonstrated for example in [12]. The problem of this approach is to find a neutral substitute, since many inputs, for example a black box in an image might already indicate a certain class in a classification problem.

Since they only highlight the important parts in one input, saliency maps are a local explanation method. The method presented in this work is a global explanation method. Furthermore, the method presented in this work can not only identify important parts of the input but also important features that are a aggregate function of these features. Many algorithms that calculate saliency maps need information on the inner workings of the estimator, such as the gradient. The method presented in this work can be applied to black-box estimators.

III. Basics and Notation

A. Machine Learning

Machine learning algorithms are algorithms that can preform tasks without being explicitly programmed but instead are presented with examples and learn from these examples. Machine learning algorithms are mostly used to train functions that preform either regression or classification tasks. For this work we focus on neural networks, even though the method presented here does not make any assumption on the estimator and can be applied to any estimator. Deep neural networks can perform regression and classification. Classification is performed by regressing the probability for every class and then using the maximum likelihood classifier, classifying the input as the most likely class. Therefore, we focus on the regression task as it is implicitly also covering the classification task.

We define a machine learning approach as a pair (T, F) of two functions. The training function T maps a set of labeled training examples $\{(S, Y_S)\}$ onto a set of weights W

$$\begin{aligned} T : \mathcal{P}(\mathbb{S} \times \mathbb{R}) &\to R^m \\ \{(S, Y_S)\} &\mapsto W \end{aligned} \tag{1}$$

and the inference function F maps an input example D and a set of weights W onto a prediction P

$$\begin{aligned} F : \mathbb{S} \times \mathbb{R}^m &\to \mathbb{R} \\ (D, W) &\mapsto P. \end{aligned} \tag{2}$$

If, for example, the machine learning method is a linear estimator, the function T is the optimization process

that maps the training examples on the optimal coefficients and the function F multiplies the coefficients with the inputs D. If the machine learning method is a k-nearest-neighbor approach, the function T is the identity, such that the set of weights is also the labeled training set and the function F is the function that identifies the k-nearest-neighbors of D in W and combines their labels into a single prediction for D.

B. Causal Inference

In this work we only use one task of causal inference. We want to determine, whether one variable is causing another variable, given all other causal relation between variables are known. To achieve this we take the following assumptions.

A structural causal model (SCM) [2] is a triplet (U, V, E) of exogenous variables U, endogenous variables V and a set of functions E. The endogenous variables often represent the observed variables, the exogenous variables represent noise in the system and the functions in E represent the causal mechanisms.

From the functions in E, a directed graph is derived. The vertices in the graph are the endogenous variables V and the graph contains a directed edge from $v_1 \in V$ to $v_2 \in V$ if v_1 is used as the input to the function defining v_2. To be able to evaluate the functions, the graph needs to be a directed acyclic graph (DAG).

Our goal is to identify, whether a certain link in the DAG exists. Even if all other links in the DAG are known, we need to employ two assumptions to be able to determine whether a specific link exists. The conditional Markov assumption states that two endogenous variables X, Y are independent given a subset $G \subset V$ if G d-separates $\{X\}$ and $\{Y\}$ in the DAG. The faithfulness assumption states that two processes are only independent given a subset $G \subset V$ if G d-separates $\{X\}$ and $\{Y\}$ in the DAG. Both of these assumptions are common in the causal inference literature. Together they give a one-to-one connection between independence and d-separation in the DAG. Even though common, these assumptions are violated in many real applications. We discuss these problems in Section VI.

IV. Method

The goal of this work is to identify whether a specific feature X is used by a black box machine learning method (T, F) for its prediction P.

We proceed by, first, showing that the machine learning approach can be modeled by an SCM and construct a possible DAG for the machine learning approach. The resulting DAG is displayed in Figure 1.

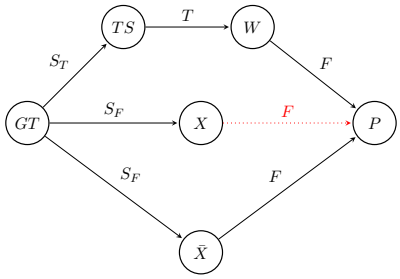


Fig. 1: The DAG of the machine learning approach. In case the Feature X is used by the black box machine learning method (T, F) to generate the prediction P, the red link exists.

The endogenous variables we look at are the ground truth GT for the prediction, the labeled training set TS, the set of weights W of the machine learning approach, the feature X for which we want to identify whether it is causing the prediction P, the set $\bar{X}$ of all features independent of X given GT, and the prediction P.

We identify the edges of the DAG from the processes used in the machine learning approach, namely the data generation and sampling processes used to create the training set TS called S_T and the features X and $\bar{X}$ called S_F and the functions F and T described in (2) and (1).

These processes are represented as edges in the DAG. The process S_T is represented as an edge from GT to TS. The process S_F is represented as an edge from GT to X and as an edge from GT to $\bar{X}$. An edge from TS to W represents T and two edges from $\bar{X}$ to P and from W to P represent F.

The only remaining question is, therefore, whether an edge between X and P exists. We know that the prediction can not cause the input feature. If a causal link between the feature X and the prediction P exists, it is directed from X to P.

Since we made the causal Markov assumption and the faithfulness assumption we know that conditional independence corresponds to d-separation in the DAG. In the graph we see that $\{GT\}$ d-separates $\{X\}$ and $\{P\}$ in the case where X does not cause P and does not d-separate $\{X\}$ and $\{P\}$ in the case where X does cause P . Hence, to identify whether the feature X is important for the prediction P of the black-box F, we check whether

$$X \perp\!\!\!\perp P | GT \quad \text{or} \quad X \not\perp\!\!\!\perp P | GT \tag{3}$$

holds.

TABLE I: Results of the experiment for all three estimators used. The p-values that indicate independence at a confidence level of 0.01 are marked in bold.

Name	MSE	p-value $\bar{A}$	p-value S
F_1	0.0026	0.5137	**0.0014**
F_2	0.0153	0.4576	**0.0091**
F_3	0.0840	**0.0052**	0.3710
F_4	0	0.05912	0.0728

V. EXPERIMENTS

A. Experimental Setup

To demonstrate that the method is able to identify features that are used by the black-box predictor for its prediction we use the following toy data set.

Let α and β be independent latent variables that influence a field $A = (a_{i,j})_{i,j\in\{1,\dots,8\}}$ of observables through

$$a_{i,j} = \alpha \cdot \varepsilon_{i,j} + \beta. \tag{4}$$

Here, $\varepsilon_{i,j} \sim \mathcal{N}(0,1)$ are independent standard normally distributed and independent of α and β. The task is to recover α from A. Note that this task is easy if we know the data creation mechanism but might be non-trivial if we are just presented with the data A and the label α.

For the feature X we test in this toy example two different features. The first is the sample mean

$$\bar{A} = \frac{1}{N}\sum_{i,j} a_{i,j} \tag{5}$$

and the second is the sample standard deviation

$$\bar{S} = \left(\frac{1}{N}\sum_{i,j}(a_{i,j} - \bar{A})^2\right)^{\frac{1}{2}}. \tag{6}$$

We compare four estimators. The first estimator F_1 is the sample standard deviation estimator

$$F_1(A) = \bar{S}. \tag{7}$$

The second estimator F_2 is a fully convolutional neural network. The network consists of three convolutional layers with 4, 16 and 64 kernels of size 2×2 and stride 2×2 followed by one convolutional layer with one kernel of size 1×1. All but the last layer have ReLU activations. We trained the neural network using Tensorflow [13] with gradient descent for 200000 steps using a learning rate of 0.00003. The third estimator F_3 is a linear estimator. To optimize F_2 and F_3 we used a training set of 20000 labeled examples. The fourth estimator F_4 is the oracle estimator that just reports the ground truth data α.

For our experiment we used an 8×8 array for A. The variables α and β are sampled from a uniform distribution. We evaluated every estimator on an identical set of 10000 examples not used for optimizing F_2 and F_3. The results can be observed in Table I. We report for every estimator the mean squared error between the true value of α and the estimated value, the confidence value "p-value $\bar{S}$" for the event

$$F(A) \perp\!\!\!\perp \bar{S}|\alpha \tag{8}$$

and the confidence value "p-value $\bar{A}$" for the event

$$F(A) \perp\!\!\!\perp \bar{A}|\alpha. \tag{9}$$

To calculate this p-values we used the fast conditional independence test presented in [14].

B. Results

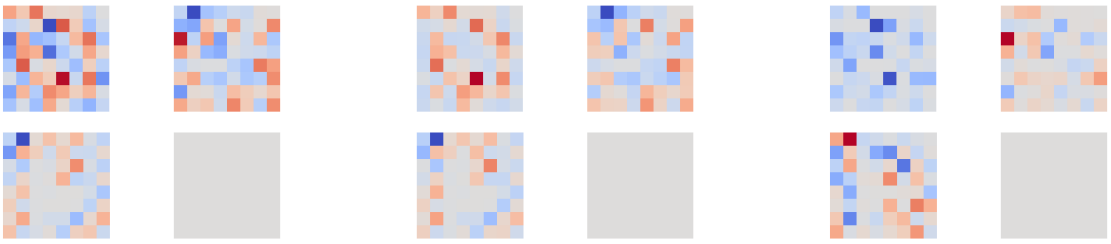
For the optimal estimator F_1, the mean squared error is small. For this estimator, we know that the only used feature is the sample standard deviation. Our method is able to correctly identify that the feature $\bar{S}$ is used by F_1 and the feature $\bar{A}$ is not used by F_1. Further for the linear estimator F_3 we know that it can not use the non-linear standard deviation estimate. We find, the linear estimator does not use the sample standard deviation, but the sample mean as a feature. It also has the highest mean squared error. For the deep neural network estimator F_2 we observe that the mean squared error is low, and the sample standard deviation is used as a feature. For the oracle F_4 we observe that none of the two features of the input is used for the estimation. This is correct, since the oracle reports the correct value independent of the input.

C. Comparison to Baseline-Methods

For comparison, in Figure 2 we display the output of the methods described in Section II. We display for all four estimators described in Section V-A the gradient of the output depending on each input (2a), the product of the gradient and the input (2b) and the change of the input if we replace one input by the mean of the other inputs (2c). In our oppinion, it is not possible to infere which features are used by the estimator from this methods. The only exception is estimator F_4 wich is can be identifyed as independent of the inputs.

VI. DISCUSSION AND OUTLOOK

The toy example showcases two use cases that we imagine for the approach presented in this work. The fist use case is to understand fail cases of estimators.



(a) The gradient of the estimator depending on the inputs.

(b) The product of gradient of the estimator and the value of the input.

(c) The difference of the output of the estimater when replacing one of the inputs by the mean of the other inputs.

Fig. 2: In every subfigure the results of one baseline method is displayed. In the top line of every subfigure shows the results for estimators F_1 and F_2, in the second line the results for estimators F_3 and F_4. Since the oracle classifier F_4 does not depend on the inputs, the results for F_4 is always zero for all inputs.

In the case of the estimator F_3, the method helps us to understand why it failed and might warn us that it will fail even more for examples with a very different mean. The second use case is to better understand the task at hand. If we have trained an estimator of high quality like F_1 or F_2, we can use the method to identify features that are relevant to solve the task. The experiment on the estimator F_4 demonstrates that conditioning on the ground truth is important and leads to more information than simply checking for independence. The method was able to correctly identify that no feature was used by the estimator F_4, even though the feature $\bar{S}$ is highly dependent with the output of the estimator.

The results on the toy data are very promising. Still, we think that a lot of future work is needed to make this method applicable to a wider range of data and to use it effectively on real-world data.

We assume that the data generation is caused by the ground truth and no further confounding, mechanism in the data generation exists. To use this method we need to be able to condition on the data generation process. We assume this drawback can be tackled using the work described in [15], [16], [17] but we leave this for future work. Further we rely on the causal Markov and the faithfulness assumption. These assuptions can be violated even in simple situations such as an XOR-gate, a trivial function or effects that cancel out. Furthermore, they are very high level assumptions that are hard to validate from other properties of the SCM. Hence, it is difficult to know whether the method of this paper can be used for a given black box predictor. Testing continuous variables for conditional independence, using only samples, is also a hard problem and often additional assumptions have to be made to solve it. Some of the drawbacks of the conditional independence test we used can be found in [14]. When using the approach presented here, one should spend time to decide on an appropriate conditional independence test based on the data used. In this work we only demonstrated a toy example with independent noise. To apply this method to real-world situations in climate science it has to handle dynamic noise and work on sparse data. We leave this for future work. A further drawback of the method is that it has to be presented with candidate features and does not generate features itself. We think, however, that this method can still help in situations in which, due to prior domain knowledge, candidate features already exist or can be generated using other methods.

Througout this work we use the notation of causality defined by the SCM. One has to be careful when linking this concept to the colloquial concept of ”causes”.

Despite these drawbacks making the method not applicable in some situations, it is still very useful in situations where it can be applied. Its main advantages are that it can provide global explanation for features that are not directly part of the input. Since climate science estimations often depend on features calculated from multiple measurements distributed in space and time and rarely single measurements cause an estimation, these advantages are needed in the field of climate science.

REFERENCES

[1] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, *et al.*, “Deep learning and process un-

derstanding for data-driven earth system science," *Nature*, vol. 566, no. 7743, p. 195, 2019.

[2] J. Pearl *et al.*, "Causal inference in statistics: An overview," *Statistics surveys*, vol. 3, pp. 96–146, 2009.

[3] J. Runge, S. Bathiany, E. Bollt, G. Camps-Valls, D. Coumou, E. Deyle, C. Glymour, M. Kretschmer, M. D. Mahecha, J. Muñoz-Marí, *et al.*, "Inferring causation from time series in earth system sciences," *Nature communications*, vol. 10, no. 1, p. 2553, 2019.

[4] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, 2017. https://distill.pub/2017/feature-visualization.

[5] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," *University of Montreal*, vol. 1341, no. 3, p. 1, 2009.

[6] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *ICLR 2014 workshop submission*, December 2013.

[7] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, pp. 818–833, Springer, 2014.

[8] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, 2017.

[9] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[10] K. R. Mopuri, U. Garg, and R. V. Babu, "Cnn fixations: an unraveling approach to visualize the discriminative image regions," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2116–2125, 2018.

[11] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep taylor decomposition," *Pattern Recognition*, vol. 65, pp. 211–222, 2017.

[12] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," *arXiv preprint arXiv:1702.04595*, 2017.

[13] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[14] K. Chalupka, P. Perona, and F. Eberhardt, "Fast conditional independence test for vector variables with large sample sizes," *arXiv preprint arXiv:1804.02747*, 2018.

[15] C. Louizos, U. Shalit, J. M. Mooij, D. Sontag, R. Zemel, and M. Welling, "Causal effect inference with deep latent-variable models," in *Advances in Neural Information Processing Systems*, pp. 6446–6456, 2017.

[16] V. T. Trifunov, M. Shadaydeh, J. Runge, V. Eyring, M. Reichstein, and J. Denzler, "Nonlinear causal link estimation under hidden confounding with an application to time-series anomaly detection," in *German Conference on Pattern Recognition (GCPR)*, 2019.

[17] V. T. Trifunov, M. Shadaydeh, J. Runge, V. Eyring, M. Reichstein, and J. Denzler, "Causal link estimation under hidden confounding in ecological time series," in *Climate Informatics Workshop*, 2019.