Connecting System Identification techniques and methods for Explainable AI

***Abstract*—**

I. Introduction

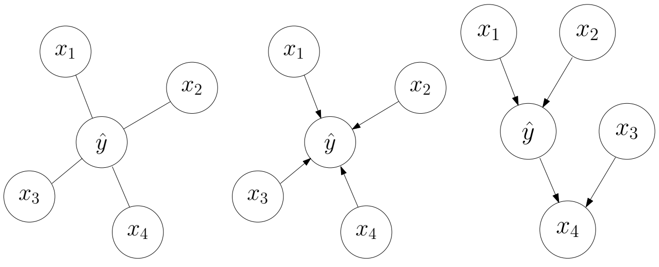
The advancements in the area of machine learning are producing algorithms capable of achieving better and better decision and predictive performance. This has naturally led to their widespread use in many areas of science and engineering. Many of these machine learning algorithms are black box in nature, meaning they are too complex for humans to interpret. This lack of opacity raises doubts about their applicability and robustness in real life scenarios. Recent works including [1], [2], and [3] argue that artificial intelligence models might not be trustworthy in certain fields. The work in [1] analyzes how the use of artificial intelligence in cybersecurity may enable attackers to exploit the learning ability of AI systems. The work in [2] and [3] explain how unnoticed bias in the training data can lead to unethical bias in the decision algorithm in medical and human resources industries, respectively. Indeed, potential hidden bias, safety and privacy considerations limit reliable use of AI models. This motivates an immense need for tools to explain AI based decisions. If we had tools to analyze AI algorithms and describe them in human interpretable ways, we may be able to increase their reliability. Another motivation for explaining AI models could come from a very different perspective than what we have considered so far; That is the reasoning behind the AI models may unveil information about the domain that was otherwise unknown to the experts. Indeed, the aforementioned reasons justify the re-emergence of explainable AI (XAI) as an active field of research [4].

The ultimate goal in XAI is to have a global explanation of an AI model. However, if the model is extremely complex and highly non-linear, a global human interpretable explanation might not accurately describe it. An alternative would be to look for local explanations, meaning, quantitatively describing why the model produced a certain output. Indeed, local explanation can be beneficial in many circumstances where AI models are being employed

The current literature categorizes explanation models based on different metrics. One such metric is generalizability. This divides explanations into global and local categories. The global explanation attempts to give insight about how the black-box model operates in general. However, this explanation may be hard to obtain if the model is highly non-linear and no information about it is available. The local explanation, on the other hand, attempts to explain one specific output of the model. Indeed, to locally explain a single data point made by the model can be of use in many areas -e.g.explaining a medical diagnosis, image/text classification. Also by explaining test samples that are sufficiently similiar we maybe able to detects patterns in the black-box model and reveal inconsistencies or gain trust of the users. Examples of methods developed for local explanation of machine learning models are Local Interpretable Model-Agnostic Explanations (LIME) TODO:CITE and expectation shapley (ES) values TODO:CITE. LIME intakes a single test sample and randomly generates a data set centered around the test sample. This data set is fed to the black-box model to get classification/prediction. Then an interpretable models is used to explain the output given the input. A case could be made for the dynamic networks when the goal is to predict a set of outputs Y from a set of inputs X.TODO: HOW TO CONNECT In the case where data belongs to the family of time series data, LIME fits a local Wiener filter TODO:CITE around the test sample. Elements of this Wiener filter can depict how the black-box model uses each feature to come to the specific prediction. Indeed, the idea is to explain how much each feature contributed to particular prediction. If the local Wiener filter component asscociated with the feature is zero, we can conclude that TODO:CITE the black-box model does not make use of the feature in order to make the prediction. It's not hard to observe that LIME retrieves the Markov blanketTODO:CITE of the prediction. The clear relation between LIME and the Markov blanket poses the question about the possibility of causal inference. The Markov blanket consists of parents, children and the spouses of the node. This set renders the predicitons independent of all other features. However, any intervention on children or the spouses does not change the probability ditstribution of the prediction. This inherent distinction motivates us to investigate ways to determine the actual causal structure in the Markov blanket. Previouse work TODO:CITE describes the use of Wiener filter to recover the causal relations that are statistically recoverable. TODO:CONTINUE

That is if we can use LIME to reconstruct the causal structure that led to the prediction made by the black-box model.

Example 1 (Possible Causal Structures Compatible With Markov Blank Figure 1(a) is a graphical representation of a Markov blanket of node $\hat{y}$ or $MB(\hat{y})$. Figure 1(b) is a potential causal structure where $\hat{y}$ has four parents. This means intervening on any of the nodes x_1, x_2, x_3, x_4 changes the probability distribution of $\hat{y}$. In contrast, Figure 1(c) illustrates another potential causal structure in which intervening on x_3 or x_4 does not change the probability distribution of $\hat{y}$.



(a) (b) (c)

Fig. 1: *(a) shows a Markov blanket of node $\hat{y}$. (b) and (c) illustrate two possible causal structures compatible with (a)*

II. Definitions Needed

Definition 1 (Markov Blanket): a Markov blanket for any given node y_i is a set of variables $MB(y_i)$ that renders y_i independent of all variables not in $MB(y_i)$.

Definition 2 (Causal Structure): TODO: WHERE TO GET IT, THE ONE FROM THE BOOK CAUSALITY SEEMS A BIT COMPLEX FOR THIS PAPER?

We denote a directed graph G as a pair (V, E) where V is the set of vertices and $E \subseteq V \times V$ is the set of directed edges. We represent a directed edge from $y_i \in V$ to $y_j \in V$ as (y_i, y_j), or $y_i \rightarrow y_j$, or $y_j \leftarrow y_i$. If $y_i \rightarrow y_j \in E$, we say that y_j is a child of y_i or, equivalently, that y_i is a parent of y_j using the notation $pa^G_{y_j}$ to denote the set of all parent of y_j. We call the nodes with common children spouses.

III. Problem Statement

Explain why a black box machine learning model made a specific prediction/classification by identifying a causal structure.

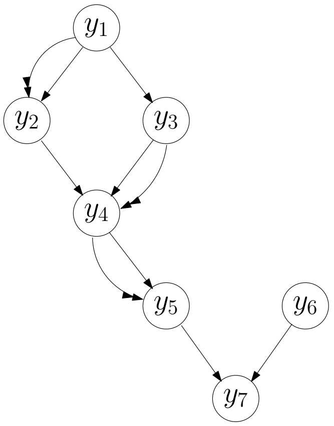


Fig. 2: *is the actual causal model of the simulation*

IV. Main Results

As mentioned previously, increasing attention is being devoted to the development of data-driven techniques for the synthesis of control algorithms. AI methods, and especially neural networks, are being leveraged in order to obtain controllers which can successfully regulate complex systems just by observing their behavior using limited a-priori knowledge. Given the inscrutable nature of most neural networks, it might often be desirable to be able to explain the basic mechanisms of such controllers before deploying them in order to guarantee their reliability and their safety. The goal of this section is to describe some fundamental tools of XAI and adapt them to the specific nature of control algorithms.

We start by considering a function v that takes a set of scalar variables $u_1, u_2, ..., u_n$ as inputs and produces the output y.

$$y = v(u_1, u_2, ..., u_n)$$

If v is a complex function, potentially implemented by training a neural network, it could be extremely challenging to obtain a complete explanation of all its decision layers and components. However, we might still be interested in obtaining an explanation of its underlying mechanisms in certain specific scenarios of interest. Indeed, a plausible and coherent local explanation would still be helpful to validate the reliability and safety of the algorithm, at least in those scenarios. Our goal is to locally explain why a specific output $y^i = v(u^i_1, u^i_2, ..., u^i_n)$ was produced. In the area of explainable AI, there are several methods to extract local explanations from complicated models CITE a survey. However, one of the most prevalent explanation methods is Local Interpretable Model-Agnostic Explanations (LIME).

A. LIME in a nutshell

LIME tries to estimate the contribution of each feature by approximating the algorithm v around the fixed input features $u^i_1, u^i_2, ..., u^i_n$ via a simpler interpretable model. In this sense, we can say that LIME is a "meta-explanation method" since its implementation depends on the specific choice of the local interpretable model. In more detail, LIME defines a class of potentially interpretable models G. Typical choices for G include linear regressions models or decision trees. To fit the model g in the neighborhood of $U^i = (u^i_1, u^i_2, ..., u^i_n)^T$, LIME randomly generates a data set Z of perturbed data points around the data point U^i. To assess how close a perturbed point $z \in Z$ is to U^i LIME considers a measure of distance noted as $\pi_i(z)$. Then, a cost function $\mathcal{L}(v, g, \pi_i)$ representing how accurate in a model g in approximating v is defined in order to facilitate the selection of an interpretable model $g \in G$. To further enhance the interpretability of these models, a common option is also to penalize the selection of models with higher complexity. To this end, LIME formulates an optimization problem of the form

$$\mathcal{E} = \arg\min_{g \in G} \sum_{z} \mathcal{L}(v(z), g(z), \pi_i(z)) + \Omega(g) \tag{1}$$

where $\Omega(g), g \in G$ is a notion of complexity (e.g. depth in decision trees or sparsity of coefficients in the linear regression models).

B. Adapting LIME to explain a black-box control algorithm

Our main technical result is the adaptation of LIME in order to explain the mechanism behind a black box control algorithm derived using data-driven methods.

In the context of control theory, we typically deal with features which are represented by time series instead of simpler scalar variables. Then, a controller can be seen as an dynamic operator v taking inputs $u_1, ...u_n$ which are time series instead of simple scalar variables.

Since we intend to apply LIME to locally explain the controller logic we need to choose an appropriate class of local interpretable models. A quite natural choice is to use Linear Time Invariant dynamic systems and consider a least square error criterion to quantify the quality of the approximation. This boils down to the approximation of the dynamic operator v around the specific input U^i using a Wiener filtering approach. Indeed, Wiener filters can be seen as the dynamic equivalent of linear regressions for time series. In terms of explainability, a multi-variate Wiener filter provides a transfer function matrix that quantitatively describes how each feature contributes to the output.

Following the philosophy of LIME, we need a perturbed set of processes defined as Z.

$$\begin{aligned} Z^i &= U^i + \epsilon \\ \bar{y}^i &= v(Z^i) \end{aligned}$$

Where $\epsilon = (\epsilon_1, \epsilon_2, ..., \epsilon_n)$ is a vector set of static random scalar processes that are mutually independent. (namely it has constant power spectral density components, $\Phi_\epsilon(z) = \Phi_\epsilon$, and its cross-power spectral density obeys $\Phi_{\epsilon_i \epsilon_j} = 0$ for $i \neq j$.) The Wiener filter estimating $\bar{y}^i$ from Z^i is given by:

$$W^i(z) = \Phi_{\bar{y}^i Z^i} \Phi_{Z^i}^{-1}$$

This realization of Wiener filter attempts to describe how the model uses the features U^i to make the prediction y^i.

C.

V. Adding causal inference to LIME

Although LIME successfully recovers a local explanation model for the function v, and it gives a sense of how each feature contributes to the output y^i, it does not gaurantee that the highlighted features actually cause a change in the output according to the model. Indeed, a high level of dependence between two variables does not necessarily imply a causal link between them CITE. Thus, inferring the existence of causal links between the features and the output, if possible, requires extra effort.

The use of Wiener filter as the interpretable model equips us with tools to potentially infer some causal properties. In particular, the work in CITE proves that the stochastic processes with non-zero components render y independent of all the other feature processes. In the context of graphical models this subset of features are called the Markov blanket. Now that we have established that no feature outside the Markov blanket can have a causal effect on the output, we can limit ourselves to this subset to determine the causal structure that function v uses to

Thus,

The next observation that we can make is that the set of features associated with the non-zero elements of the Wiener filter form the Markov blanket of the output.

References

[1] M. Taddeo, T. McCutcheon, and L. Floridi, "Trusting artificial intelligence in cybersecurity is a double-edged sword," *Nature Machine Intelligence*, vol. 1, no. 12, pp. 557–560, 2019.

[2] D. F. Mujtaba and N. R. Mahapatra, "Ethical considerations in ai-based recruitment," in *2019 IEEE International Symposium on Technology and Society (ISTAS)*. IEEE, 2019, pp. 1–7.

[3] V. N. O'Reilly-Shah, K. R. Gentry, A. M. Walters, J. Zivot, C. T. Anderson, and P. J. Tighe, "Bias and ethical considerations in machine learning and the automation of perioperative risk assessment," *British Journal of Anaesthesia*, vol. 125, no. 6, pp. 843–846, 2020.

[4] S. R. Islam, W. Eberle, S. K. Ghafoor, and M. Ahmed, "Explainable artificial intelligence approaches: A survey," *arXiv preprint arXiv:2101.09429*, 2021.