Reasoning about algorithmic opacity

Ekaterina Kubyshkina[1], Mattia Petrolo[2]

[1]University of Milan, Philosophy Department, Via Festa del Perdono, 7 20122, Milan, Italy
[2]Federal University of ABC, Alameda da Universidade, s/n, São Bernardo do Campo, SP 09606-045, Brazil

Abstract

A recurring problem discussed in explainable AI is the so-called epistemic opacity problem, that is, a problem about the epistemic accessibility and reliability of algorithms. In the present work, we provide an original epistemological characterization of the opacity of algorithms based on a tripartite analysis of their components. Against this background, we introduce a formal framework by modifying the neighborhood semantics for evidence logic introduced in [1]. This setting allows one to reason about an agent's epistemic attitudes toward an algorithm and investigate what are the conditions that should be met to achieve epistemic transparency.

Keywords

Transparent AI, epistemic opacity, epistemic logic, evidence models, neighborhood semantics

1. Introduction

The explosion in the use of computational algorithms in several domains of human life prompted the development of explainable AI and, more in general, of what can be labelled as a *human-centered approach* to algorithms to help shed some light on the nature of AI models. From this perspective, as Seaver puts it, "it is not the algorithm, narrowly defined, that has sociocultural effects, but *algorithmic systems* - intricate, dynamic arrangements of people and code" ([2], pp. 418-419). A recurring problem discussed in this approach, in a form or another, is the so-called epistemic opacity problem, that is, a problem about the epistemic accessibility and reliability of algorithms. In the present work, our aim is to provide an original epistemological and logical characterization of the epistemic opacity of algorithms, in order to investigate under which conditions this form of opacity can be eliminated.

2. A definition of epistemic opacity

To characterize the epistemic opacity of algorithms, we follow the methodology proposed by Durán and Formanek [3] and adapt Humphreys' definition of epistemically opaque process:

> [A] process is epistemically opaque relative to a cognitive agent X at time t just in case X does not know at t all of the epistemically relevant elements of the process ([4], p. 618).

1st Workshop on Bias, Ethical AI, Explainability and the role of Logic and Logic Programming, BEWARE-22, co-located with AIxIA 2022, University of Udine, Udine, Italy, 2022
ekaterina.kubyshkina@unimi.it (E. Kubyshkina); mattia.petrolo@ufabc.edu.br (M. Petrolo)

This characterization of opacity crucially relies on the fact that an agent "X does not *know*", which, in turn, presupposes an account of what knowledge is. However, unfortunately, Humphreys leaves this question open. Traditional epistemology takes knowledge as corresponding to justified true belief. This analysis has been challenged by Gettier's famous counterexample (see [5]), which prevents one to consider luck-dependent cases as cases of genuine knowledge. To avoid this problem, our characterization of opacity must take special care in spelling out the nature of the justificatory component involved in the analysis. Moreover, in order to specify what are the "epistemically relevant elements" of algorithms, the definition has to take into account their specific structure. Cormen et al. [6] describe an algorithm as follows: "Informally, an algorithm is any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output" ([6], p. 5). Although informal and sketchy, this description highlights three fundamental elements of an algorithm: its input, procedure, and output. We argue that a sound characterization of epistemic opacity (and epistemic transparency) for algorithms has to take these elements into account. Our proposal is as follows:

Definition 2.1 (Epistemological)**.** An algorithm is *epistemically opaque* relative to an epistemic agent A at time t just in case at t, A does not have
an epistemic justification for I,
or an epistemic justification for P,
or an epistemic justification for O;
where I, P, O express the algorithm's input, procedure, and output, respectively.

One important feature of the previous definition is that the components I, P, and O of the algorithm are, at least in principle, independent. As a consequence, the lack of epistemic justification for any component constitutes a sufficient condition for epistemic opacity. Let us clarify the definition via an example of a specific facial recognition algorithm used in AI. Convolutional neural networks (CNN, see [7]) are a biologically-inspired class of neural networks used in facial recognition. CNN consist of three layers: an input layer, an output layer, and several hidden layers (e.g., convolutional layers, pooling layers, fully connected layers). Following Definition 2.1, a CNN can be opaque for three different reasons. First, an agent using the model might not know the set of images used to train the neural network. This *input opacity* is external to the procedure of the algorithm and it depends on some choices made by the algorithm's designers, which are not necessarily accessible to the user. Thus, input opacity is not about a single image inserted by the user, but rather about the relationship established between this image and the dataset on which the algorithm is trained. Second, the agent using the model might not have epistemic access to the procedures of the hidden layers. For instance, she misses the information about what a convolution operation is. This *procedure opacity* is internal to the algorithm and presupposes some form of epistemic access to the inner working of the algorithm. Finally, the epistemic agent using the algorithm might notice that the output does not fit with her current set of beliefs. For instance, she knows that two faces are the same, despite the output of the CNN claims otherwise. This *output opacity* is external to the procedure of the algorithm and it represents for the user a way to compare the output with her previously acquired beliefs. In principle, the three conditions can occur separately, but, in most of real-world algorithms, these forms of opacity are entangled, thus raising the

complexity of the epistemic opacity problem. On the basis of this characterization, in the vast majority of cases, the algorithms with which we interact on a daily basis are epistemically opaque. In what follows, we introduce a formal framework to reason about an agent's epistemic attitudes towards opaque algorithms and investigate what are the conditions that should be met to achieve epistemic transparency.

3. A formal framework

Definition 2.1 can be considered as a tentative epistemological characterization of epistemic opacity. However, to reason formally about the epistemic attitudes of an agent toward opacity, one needs to recast that definition in logical terms. To do so, we will borrow some tools from the toolkit of epistemic logic and match each component of Definition 2.1 with an epistemic modality. Roughly, we consider that the fact that an agent has an epistemic justification for I can be logically represented by $K\phi$, that is, an agent "knows the input ϕ" of the algorithm. The fact that an agent has an epistemic justification for P can be seen as $\Box\phi$, that is, the agent "has an evidence for the procedure ϕ". Finally, the fact that an agent has an epistemic justification for O can be considered as $B\phi$ that is, the agent "believes in the output ϕ". Let us now present more carefully this formal framework.

The semantics we are proposing is a modification of a neighborhood semantics for evidence logic provided by van Benthem et al. [1]. The main difference of our proposal from [1] is that we need to fix three separate domains: for inputs, for procedures, and for outputs. For inputs we fix a domain consisting of variables $a, b, c, ... \in At_{in}$. These variables denote some data inserted into the algorithm. Practically, the data can be in a form of sentences, pictures, diagrams etc. For this reason $a, b, c...$ do not designate only propositions. For procedures we fix a domain consisting of variables $\alpha, \beta, \gamma, ... \in At_{pr}$. These variables denote procedures used by the algorithm. As procedures are not just propositions, $\alpha, \beta, \gamma, ...$ are not just propositional variables. For the outputs we fix a domain of propositional variables $p, q, r, ... \in At_{out}$ by assuming that the outputs of the algorithms always take propositional form.

The language $\mathscr{L}$ is defined as follows:

$$\phi := p \mid \alpha \mid a \mid \neg\phi \mid \phi \wedge \phi \mid B\phi \mid \Box\phi \mid K\phi$$

where $p \in At_{out}$, $\alpha \in At_{pr}$, $a \in At_{in}$, and ϕ in $B\phi$ is defined on the domain At_{out}, ϕ in $\Box\phi$ is defined on the domain At_{pr}, ϕ in $K\phi$ is defined on the domain At_{in}.

The intended interpretation of $B\phi$ is "the agent believes in the output ϕ," where ϕ stands for a proposition. The interpretation of $\Box\phi$ is "the agent has evidence for procedure ϕ." By "having an evidence for a procedure" we mean that an agent has an understanding of a particular way of producing an output based on a given input. The interpretation of $K\phi$ is "the agent knows the data ϕ," where by "knowing the data" we mean that the agent is aware of the inputs of the algorithm. From this perspective, we are not dealing with propositional factive knowledge in this case. For instance, an agent may know the data used by the algorithm as an input even if these data are incorrect. In what follows, we indicate by At the union of At_{out}, At_{pr}, and At_{in}, and we add a superscript on variables p^x, α^x, a^x in order to mark them, respectively, as the output, the procedure, and the input of the same algorithm x.

In order to interpret the language $\mathscr{L}$ we use the *evidence models* introduced by van Benthem et al. [1].

Definition 3.1. An evidence model is a tuple $\mathscr{M} = \langle W, E, V \rangle$, where W is a non-empty set of worlds, $E \subseteq W \times \mathfrak{p}(W)$ is an evidence relation, $V : At \to \mathfrak{p}(W)$ is a valuation function. We write $E(w)$ for the set $\{X | wEX\}$. Two constraints are imposted on the evidence sets: For each $w \in W$, $\emptyset \notin E(w)$ and $W \in E(w)$.

Definition 3.2. A w-scenario is a maximal collection $\mathscr{X} \subseteq E(w)$ that has the finite intersection property: for each finite subfamily $\{X_1, ..., X_n\} \subseteq \mathscr{X}$, $\cap_{1 \leq i \leq n} X_i \neq \emptyset$.

Definition 3.3. Let $\mathscr{M} = \langle W, E, V \rangle$ be an evidence model. Truth of a formula $\phi \in \mathscr{L}$ is defined as follows:

- $\mathscr{M}, w \vDash p$ iff $w \in V(p)$;
- $\mathscr{M}, w \vDash \alpha$ iff $w \in V(\alpha)$;
- $\mathscr{M}, w \vDash a$ iff $w \in V(a)$;
- $\mathscr{M}, w \vDash \neg\phi$ iff $\mathscr{M}, w \nvDash \phi$;
- $\mathscr{M}, w \vDash \phi \wedge \psi$ iff $\mathscr{M}, w \vDash \phi$ and $\mathscr{M}, w \vDash \psi$;
- $\mathscr{M}, w \vDash \Box\phi$ iff there exists X such that wEX and for all $v \in X$, $\mathscr{M}, v \vDash \phi$;
- $\mathscr{M}, w \vDash B\phi$ iff for each w-scenario $\mathscr{X}$ and for all $v \in \cap\mathscr{X}$, $\mathscr{M}, v \vDash \phi$;
- $\mathscr{M}, w \vDash K\phi$ iff for all $v \in W$, $\mathscr{M}, v \vDash \phi$.

  The satisfiability and validity are defined as usual.

The main peculiarity of our approach lies in distinguishing three types of domains for variables and limiting the application of each modal operator by its corresponding domain. Technically, we have defined the same function for all the three types of variables, and the evidence sets are defined so that they can contain any type of variables. However, conceptually, satisfiability of each type of variables in a world represents different situations. In particular, $\mathscr{M}, w \vDash p$ means that proposition p is true in a world w, which is standard. By $\mathscr{M}, w \vDash \alpha$ we mean that the procedure α belongs to the world w. This, in turn, by definition of $\neg$, means that $\mathscr{M}, w \vDash \neg\alpha$ should be interpreted as the fact that the procedure α does not belong to the world w. Similarly, $\mathscr{M}, w \vDash a$ means that a belongs to the world w, and $\mathscr{M}, w \vDash \neg a$ means that a does not belong to the world w. These considerations are useful for understanding the definitions of B, $\Box$, and K. In particular, $\mathscr{M}, w \vDash B\phi$ means that after considering all the evidences for and against ϕ, the truth of ϕ is consistent with these evidences. The condition for $\mathscr{M}, w \vDash \Box\phi$ states that an agent has an evidence for a procedure ϕ iff ϕ is present in some evidence set available in w. Notice that it is possible that an agent has an evidence both for ϕ and $\neg\phi$. This is in line with our informal reading, because an agent can have evidences for a procedure being applicable in a certain context, but inapplicable in some other. Finally, $\mathscr{M}, w \vDash K\phi$ means that the data is present in all worlds considered by the agent, i.e., the agent has full access to the data in all contexts.

On the basis of the previous definitions, we are able to define opaque algorithms semantically. We do not need to define an algorithm in our semantics, but we take it as a system x containing all inputs, procedures, and outputs, associated with x. Following Definition 2.1, an opaque

algorithm (Ox) is one for which the agent lacks a justification for at least of one of its components. We can now rephrase this definition in logical terms.

Definition 3.4 (Logical)**.** An algorithm is *epistemically opaque* relative to an epistemic agent in a world $w \in \mathscr{M}$ if:

$\mathscr{M}, w \models Ox$ iff $\mathscr{M}, w \models \neg K\phi_1^x \vee \neg\Box\phi_2^x \vee \neg B\phi_3^x$.

Now we are able to provide semantic models for representing transparent and opaque algorithms.

Example 3.1 (Transparent algorithm)**.** Let $\mathscr{M} = \langle W, E, V\rangle$ such that $W = \{w, v_1, v_2, v_3\}$, $E(w) = \{\{w, v_1, v_2, v_3\}\}$, and $V(a^x) = V(\alpha^x) = V(p^x) = W$. Clearly, in this model we have $\mathscr{M}, w \models Ka^x$, $\mathscr{M}, w \models \Box\alpha^x$, and $\mathscr{M}, w \models Bp^x$. Thus, $\mathscr{M}, w \models \neg Ox$.

Example 3.2 (Opaque algorithm - 1)**.** In this example we provide a model for an algorithm, the opaqueness of which is due to the lack of epistemic justification for the input, in presence of epistemic justifications for the procedure and the output. Let $\mathscr{M} = \langle W, E, V\rangle$ such that $W = \{w, v_1, v_2, v_3\}$, $E(w) = \{\{w, v_1, v_2, v_3\}, \{w, v_1, v_2\}\}$, and $V(a^x) = \{w, v_1, v_3\}$, $V(\alpha^x) = \{w, v_1, v_2\}$, $V(p^x) = \{w, v_1, v_2, v_3\}$. In this model we have $\mathscr{M}, w \models \neg Ka^x$, because $\mathscr{M}, v^2 \models \neg a^x$; and we have $\mathscr{M}, w \models \Box\alpha^x$, $\mathscr{M}, w \models Bp^x$. Thus, $\mathscr{M}, w \models Ox$.

Example 3.3 (Opaque algorithm - 2)**.** Now we model a situation in which an algorithm is opaque due to the lack of epistemic justification for the procedure, in presence of epistemic justification of the input and of the output. Let $\mathscr{M} = \langle W, E, V\rangle$ such that $W = \{w, v_1, v_2, v_3\}$, $E(w) = \{\{w, v_1, v_2, v_3\}, \{w, v_1, v_2\}\}$, and $V(a^x) = \{w, v_1, v_2, v_3\}$, $V(\alpha^x) = \{v_1\}$, $V(p^x) = \{w, v_1, v_2, v_3\}$. In this model, we have $\mathscr{M}, w \models \neg\Box\alpha^x$, because for all X such that wEX there exist $v \in X$ such that $\mathscr{M} \not\models \alpha^x$; and we have $\mathscr{M}, w \models Ka^x$, $\mathscr{M}, w \models Bp^x$. Thus, $\mathscr{M}, w \models Ox$.

Example 3.4 (Opaque algorithm - 3)**.** Here we consider a model, in which an algorithm is opaque because the agent lacks an epistemic justification for the output, in the presence of the justifications both for the input and the procedure. Let $\mathscr{M} = \langle W, E, V\rangle$ such that $W = \{w, v_1, v_2, v_3\}$, $E(w) = \{\{w, v_1, v_2, v_3\}, \{w, v_1, v_2\}\}$, and $V(a^x) = \{w, v_1, v_2, v_3\}$, $V(\alpha^x) = \{w, v_1, v_2\}$, $V(p^x) = \{w, v_1, v_3\}$. We have $\mathscr{M}, w \models \neg Bp^x$, because for $v_2 \in \cap\mathscr{X}$, $\mathscr{M}, v_2 \not\models p^x$; and we have $\mathscr{M}, w \models Ka^x$, $\mathscr{M}, w \models \Box\alpha^x$. Thus, $\mathscr{M}, w \models Ox$.

4. Conclusion and future work

We provided an original epistemological definition of algorithmic opacity based on a tripartite analysis of algorithms. On the basis of this definition, we introduced a formal framework that allows one to analyze the epistemic attitudes of an agent towards a possibly opaque algorithm. The transition from the epistemological to the formal framework is made by respecting the tripartite structure of I, P, and O and by attributing to them the K, $\Box$, and B modality, respectively. Let us call this analysis the *IPO model* of algorithmic opacity. In future work, we aim at deepening the IPO model, both from an epistemological and formal perspective. Regarding the former, in the literature on algorithmic opacity, it is often pointed out that one of the major difficulties in analyzing opacity is its multilayered nature. Some authors have proposed taxonomies for

different forms of epistemic opacity. For instance, Burrell [8] distinguishes between intentional, illiterate, and intrinsic opacity. From this perspective, we intend to compare the epistemological definition we introduced with the forms of opacity analyzed in the literature, in order to understand whether our definition is general enough to encompass all possible forms of opacity. From a formal point of view, we have adapted the evidence models to provide a general semantic framework to reason about an agent's epistemic attitudes toward an algorithm. The next natural step in this investigation is to introduce a logical system for reasoning about opacity and prove its completeness with respect to the evidence models. Moreover, recently, other approaches for dealing with the notion of evidence were proposed, for instance by Carnielli and Rodrigues [9] and Artemov [10]. Carnielli and Rodrigues [9] provide an interpretation of paraconsistent and paracomplete logics in terms of evidence. Even though this reading permits one to deal with possibly inconsistent evidence, which is also possible by using the semantics we adopted, the relation between evidence and justification is less straightforward in this logical framework. For this reason, we consider our approach more promising for the aims of the current work. Artemov [10] introduced a logic of justification by supplementing the standard modalities of epistemic and doxastic logic with explicit terms, which provide the reason for believing in a proposition. It seems that the IPO model can also be represented in this framework. We leave the task of adapting the logic of justification to the analysis of opaque algorithms for future investigations.

Acknowledgments

The authors acknowledge the support of the Project PRIN2020 BRIO - Bias, Risk and Opacity in AI (2020SSKZ7R) awarded by the Italian Ministry of University and Research (MUR). The research of Ekaterina Kubyshkina is funded under the "Foundations of Fair and Trustworthy AI" Project of the University of Milan. Mattia Petrolo gratefully acknowledges the support of the French National Research Agency (ANR) through the Project ANR-20-CE27-0004.

References

[1] J. van Benthem, D. Fernández-Duques, E. Pacuit, Evidence logic: A new look at neighborhood structures, Advances in Modal Logic (2012).
[2] N. Seaver, Knowing algorithms, in: J. Vertesi, D. Ribes (Eds.), Digital STS: A Field Guide, Princeton University Press, 2019, pp. 412–422.
[3] J. Durán, N. Formanek, Grounds for trust: Essential epistemic opacity and computational reliabilism, Minds and Machines 28 (2018) 645–666.
[4] P. W. Humphreys, The philosophical novelty of computer simulation methods, Synthese 169 (2009) 615–626.
[5] E. Gettier, Is justified true belief knowledge?, Analysis 23 (1963) 121–123.
[6] T. Cormen, C. E. Leiserson, R. Rivest, C. Stein, Introduction to Algorithms, Cambridge: MIT Press, 2009.
[7] Y. LeCun, Y. Bengio, Convolutional networks for images, speech, and time series, in:

M. Arbib (Ed.), The handbook of brain theory and neural networks, 2nd ed., The MIT press, 1998, pp. 255–258.

[8] J. Burrell, How the machine 'thinks': Understanding opacity in machine learning algorithms, Big Data & Society 2 (2016) 1–12.

[9] W. Carnielli, A. Rodrigues, An epistemic approach to paraconsistency: a logic of evidence and truth, Synthese 196 (2019) 3789–3813.

[10] S. Artemov, The logic of justification, The Review of Symbolic Logic 1 (2008) 477–513.