BNAIC/Benelearn Conference on AI and Machine Learning, Delft, The Netherlands, 2023

Defining and Evaluating the Optimal Degree of Abstraction in Explanations with Kolmogorov complexity

Jan Lemeire[1,2][0000−0002−2106−448X] and Stefan Buijsman[3][0000−0002−0004−0681]

[1] Dept. of Industrial Sciences (INDI), Vrije Universiteit Brussel (VUB), Pleinlaan 2, B-1050 Brussels, Belgium
[2] Dept. of Electronics and Informatics (ETRO), Vrije Universiteit Brussel (VUB), Pleinlaan 2, B-1050 Brussels, Belgium
[3] TU Delft, Section of Ethics and Philosophy of Technology, Jaffalaan 5, 2628 BX Delft, NL
jan.lemeire@vub.be

Abstract. What variables should be used to get explanations (of AI systems) that are easily interpretable? The challenge to find the right degree of abstraction in explanations, also called the 'variables problem', has been actively discussed in the philosophy of science. The challenge is striking the right balance between specificity and generality. Concepts such as proportionality and exhaustivity are investigated and discussed. We propose a new and formal definition based on Kolmogorov complexity and argue that this corresponds to our intuitions about the right level of abstraction. First, we require that variables are appropriately uniform, so that they cannot be decomposed into less abstract variables without increasing the Kolmogorov complexity. Next, uniform variables are optimal for an explanation if they can compose its domain without increasing its Kolmogorov complexity. For this, the concepts K-decomposability and K-composability of sets are defined. Explanations of a certain instance should encompass a maximal set of instances without being K-decomposable. Although Kolmogorov complexity is uncomputable and depends on the choice of programming language, we show that it can be used effectively to evaluate and reason about explanations, such as in the evaluation of XAI methods.

Keywords: Explainability · Explainable AI · Kolmogorov complexity

1 Introduction

How do we best explain a particular outcome of a binary function in terms of the properties of the input? One of the challenges in answering this question is finding the right variables to use in these explanations. Intuitively, we prefer an explanation that is not too specific, nor too abstract. Consider a Convolutional Neural Network (CNN) that is trained to recognize coffee in images and assume the network successfully recognizes coffee in all the images shown in Figure 1. An explanation for the identification of coffee in image (a) is the dark brown color and the foam. But if image (b) also leads to a positive identification, the property 'dark brown' is too specific; 'brown' is sufficient and seems to better capture the behaviour of the network. Similarly, image (c) is linked

Fig. 1. Images of coffee that are correctly identified by a trained Convolutional Neural Network.

to coffee by the shape of the cup and the steam, but a very specific description of the cup may lead us to believe that the system generalizes less than it does if the more abstract cup in image (d) is also classified as a coffee cup.This can be observed again in image (e) where the 'flower pattern' in the foam may not be necessary for identification if image (f) is also classified correctly. Vice versa, 'rounded shapes' may be too general a variable for the explanation, even if it matches these examples. The final image illustrates this again, where one may use 'breakfast' is too abstract for image (g). An explanation is preferred that points to the actual properties of the image that lead to the identification of coffee. For image (g) this is 'a croissant, newspaper and a cup'.

However, it is a challenge to specify formally what this optimal degree of abstraction is and on which concepts, called *variables* in the philosophical literature, an explanation should be based. This is a general problem for theories of explanation [5, 7], but one that reasserts itself in the field of XAI [6]. Especially in the philosophy of science there has been earlier work on precisely this question, which we survey in section 2. Our aim in this paper is to build upon this work by giving a formal definition of this degree of abstraction using Kolmogorov complexity theory. We therefore first introduce Kolmogorov complexity theory in section 3. In section 4 we point towards a failure of current approaches and provide an alternative formal specification for the optimal degree of abstraction of a variable in an explanation. While this will not give us a way to find the absolute best set of concepts to use, it does provide a way to compare two competing explanations along the dimension of the abstractness of their variables. We then show in section 5 how this definition applies and resolves the current problems. In the final section we illustrate how this theoretical discussion applies to XAI methods for explanations.

With the context of XAI as part of the motivation for formalizing the discussion on abstraction, we will consider the binary classifier b which outputs 0 or 1 for each input $x \in X$. x is a multi-dimensional feature vector. The set of all inputs for which b outputs a 1 is called the *positive subset*, which we denote with S_b. We return to this in section 3, but b is also called the *indicator function* for set S_b. We then use the following definition of an explanation as applied to black box algorithm b: "An explanation of output y_1 of

b, resulting from input x_1 is: a generalization G where $G(x_1) = b(x_1) \pm \delta$, with δ a chosen minimum accuracy of G such that furthermore, there is at least one set of inputs x_2 where $G(x_2) = b(x_2) \pm \delta$ and $b(x_1) \neq b(x_2)$" [6, p.567], based on [18]. An explanation of an instance is based on the properties of the image that ensured the outcome. Whether the property is present in the image determines the outcome of b. A property thus corresponds to a set of images and an explanation G to a subset S_G of the positive subset S_b. The problem for this paper is then to determine what the optimal variables are in the function G and how large the subset S_G should be. For this, we will turn to algorithmic information theory. First, however, we will discuss related (philosophical) work on this issue.

2 Related work

There is a wide-ranging literature on explanations of AI systems [1]. Methods showing how important different features were for the output [15, 13] are one option, as are methods extracting rules (e.g. decision trees) to describe the (local) behaviour of the AI system and counterfactuals showing what should be changed to the input to achieve the desired output [11]. And while there is still disagreement about how we should define explanations, both in the computer science literature [9] and in the philosophical literature on explanation [3], the overall goal on all of these definitions is to let recipients of explanations better understand the AI system.

In all of these cases, too, explanations require the use of variables: either the input variables of the system, or a set of (often more abstract) variables that are closer to the variables humans are used to working with. Examples of the latter are so-called concept-based explainability methods, such as Concept Activation Vectors [8, 20] which attempt to extract (some of) the patterns that a convolutional neural network uses to arrive at the output classification. Alternative methods use crowd workers to attribute concepts to highlighted regions in images [2, 4], thus abstracting from highlighted individual pixels to more abstract concepts. This makes explanations not only more interpretable, as humans are more used to reasoning with concepts such as chairs and tables than we are with sets of pixel values. It also makes explanations more general, as more abstract variables typically cover a wider set of cases.

This is important, as a common standard for the quality of an explanation is how general the explanation is [6]. In other words, “powerful explanations should, just like any predictor, generalize as much as possible" [11, p.36]. However, finding the point where an explanation has generalized *as much as possible* is difficult. The ‘variables problem’ [7, 17] in the philosophy of explanation shows the challenge of identifying the right degree of abstraction for explanations. To illustrate with an example commonly used in philosophical literature, there is an intuitive sense that of the following three explanations the second is the best, being neither too specific nor too general:

(1) The pigeon pecked (rather than only looked) because it was presented with a scarlet stimulus (rather than some other stimulus)
(2) The pigeon pecked (rather than only looked) because it was presented with a red stimulus (rather than some other stimulus)

(3) The pigeon pecked (rather than only looked) because it was presented with something stimulating (rather than some other thing)

Specifying why this is so is non-trivial, but by now two approaches can be found. [5] suggests that we opt for the most abstract variables that are still specific, where abstraction and specificity are defined as follows:

> An explanation with explanans variable(s) e_1 is more abstract than an explanation with explanans variable(s) e_2 when the actual value of e_1 is implied by the actual value of e_2, but not vice versa
>
> An explanation with explanans variable(s) e_1 is more specific than an explanation with explanans variable(s) e_2 when e_2 is a function f of e_1 and other variables e_3, $\ldots e_n$ such that for $e_1 = e_{1,A}$ neither e_1 nor $G(e_2) = G(f(e_1, e_3 \ldots e_n))$ change value if the variables e_3, $\ldots e_n$ are varied.

These definitions apply as follows: using *red* leads to a better explanation than using *scarlet* because it is more abstract (if *scarlet* = 1 then *red* = 1, but not vice versa) without being more specific. On the other hand, using *red* leads to a better explanation than using *something stimulating* because it is more specific (*something stimulating* can be seen as a function $f(red, food, tickle) = red \vee food \vee tickle$ where the value of f remains the same as long as *red* = 1).

[19] takes a slightly different approach to the same problem, stating that a requirement of proportionality instead motivates the choice of variable. This principle of *proportionality* states that "ther things being equal, we should prefer those causal claims/explanations that more fully represent or exhibit those patterns of dependence that hold" [19, p.247]. It then functions as follows: using *scarlet* suggests the following relation: if *scarlet* = 1 then *peck* = 1, if *scarlet* = 0 then *peck* = 0. The latter part is false, as the bird will also peck when other shades of red are presented. Hence, the explanation that if *red* = 1 then *peck* = 1, if *red* = 0 then *peck* = 0 is better.

3 The Kolmogorov complexity of functions

To introduce Kolmogorov complexity, we consider first how we can describe an indicator function of a set. If a set contains only *random* elements then a description must rely on an enumeration of all elements. As they are random, they will have, in general, no properties in common. In most cases, however, we are interested in sets that mean something, of which the elements have properties in common on which the indicator function can be built. Then, the implementation of the indicator function will become shorter than a literal enumeration. This can be formalized by algorithmic information or 'Kolmogorov complexity', a concept put forward as an objective measure of complexity.

3.1 Definition of Kolmogorov complexity

First we define the Kolmogorov complexity (KC) of a single object:

Definition 1. *For a binary sequence $x \in \{0,1\}^*$, the algorithmic information $K(x)$ (or 'Kolmogorov complexity') is defined as the length of the shortest program on a universal Turing machine that generates x and then stops:*

$$K(x) = \min_{p:\mathcal{U}(p)=x} l(p) \tag{1}$$

with $\mathcal{U}$ a universal computer, and $l(.)$ the length in bits of a binary sequence.

The shortest program is denoted with p_x^*.

To illustrate this definition, consider the following two sequences of 1000 bits:

- 01111000011001100111 ... 00001111100100011101
- 00010001000100010001 ... 00010001000100010001

The first string is random, while the second repeats "0001". $K(x)$ is maximal for the random string, namely 1000 bits. The shortest program literally encodes the string. The second string can be described by program `REPEAT 250 TIMES "0001"` and needs far fewer bits. The program exploits the 'regularities' (patterns) of the string to *compress* its description. It is these regularities that make up the meaningful information we are interested in. This same idea can then be applied to indicator functions:

Definition 2. *The **Kolmogorov complexity of a binary function** b that takes as argument $x \in X$ and returns a 0 or a 1, is defined as the length of the shortest program p^* that when executed by a universal Turing machine together with any argument $x \in X$ returns the same output as $b(x)$: $\mathcal{U}(p^*, x) = b(x)$. The shortest program is denoted as p_b^*,*

Definition 3. *The **Kolmogorov complexity of a set** $S \subseteq X$ is defined as the Kolmogorov complexity of the indicator function of S.*

3.2 Limitations and practical use of Kolmogorov complexity

There are two problems to apply Kolmogorov complexity to practical problems [10]. First, Kolmogorov complexity is not computable. It is proven that there is no algorithm that given a bitstring will output the length of the shortest program and halts. For a lot of cases, however, the shortest program is indisputable, as will be shown in the discussed examples. Still, for more intricate programs it is not trivial, as for example in the case of neural networks trained to detect objects – which quickly use millions of parameters. Instead of trying to identify the absolute shortest implementation (and with that the absolute best concept to use in the explanation), we will therefore use the definitions to *compare and validate implementations* in the same way as explanations are compared in philosophical literature.

Second, Kolmogorov complexity depends on the choice of programming language up to a constant. Since one programming language can be translated into another one with a program of length C, the difference of describing x in both languages, can be maximally be C. Therefore, theorems often have to incorporate this constant [12]. This

can be seen in the additivity rule (which we need later) for the joint Kolmogorov complexity has the following formulation:

$$K(x,y) \stackrel{+}{=} K(x) + K(y|p_x^*)\,, \tag{2}$$

where $K(y|p_x^*)$ denotes the conditional Kolmogorov complexity of y, given the shortest program p_x^* of x. As usual in algorithmic information theory, $\stackrel{+}{=}$ denotes equality up to a constant that is independent of the string x, but does depend on the Turing machine.

4 Abstraction and undecomposable concepts

The underlying idea behind the formal definitions that we introduce below is that variables should be both general and, at the same time, undecomposable. In other words, the variable should track a single property that can be tested for in a *canonical* manner. To see how this differs from the abstraction - specificity suggestion of [5], it helps to consider the following two variations on the pigeon example.

4.1 Illustration of the challenge

First, imagine a scenario where a pigeon pecks at both red and yellow stimuli. In this case, we could use an explanation with the two colour variables separately, arriving at the explanation:

(4) The pigeon pecked (rather than only looked) because it was presented with either a red or a yellow stimulus (rather than some other stimulus)

Or we could introduce a new, more abstract variable *redow* $= red \vee yellow$ and use the following explanation:

(5) The pigeon pecked (rather than only looked) because it was presented with a redow stimulus (rather than some other stimulus)

If we compare these two explanations there seems to be a clear preference for (4), where no new abstract variable is introduced to cover the two cases in one go. However, judging by the criteria of [5] we should in fact prefer (5). The variable is more abstract (*yellow* = 1 implies *redow* = 1 and *red* = 1 implies *redow* = 1) but not more specific (*red* changes values if the value of *yellow* is changed and vice versa).

To get to the idea that it is the uniformity of the concept, consider a second variation on the same example. Here, there is a range of different colours that a pigeon responds to. What they all have in common is that they are bright colours, such as red, orange and yellow. So, we again have two options for an explanation. Either we use the more abstract variable *bright colour* or we use a disjunction of less abstract variables:

(6) The pigeon pecked (rather than only looked) because it was presented with a red or an orange or a yellow stimulus (rather than some other stimulus)

(7) The pigeon pecked (rather than only looked) because it was presented with a bright coloured stimulus (rather than some other stimulus)

Both [5] and [19] will correctly classify (7) as the better explanation. Still, it seems that the abstract concept in (7) is preferable whereas the abstract concept in (5) is not, even though in principle both are specifiable as disjunctions of incompatible colour concepts. So what is the difference? In our view, it is that there is a separate, undecomposable way to define *bright colour*. Specifically, colours can be defined using the HSV colour space (`https://en.wikipedia.org/wiki/HSL_and_HSV`), where the V-component defines the brightness using a single numerical value. There is no such unifying measure for *redow*, which could be why we find this a less illuminating concept to use.

4.2 Formal definition of uniformity

Formalizing this idea of having a unifying measure and undecomposable definition available for a concept we can appeal to Kolmogorov complexity to define when a concept meets this requirement. To make this translation we have to interpret concepts as sets, where the set has as members every element to which the concept applies. The question of whether the corresponding concept is appropriately uniform can then be approached in terms of the Kolmogorov complexity of the description of the set:

Definition 4. *A set S is K-decomposable if there exist different and non-empty subsets S_1 and S_2 such that:*

- $S = S_1 \cup S_2$, *and*
- $K(S) \stackrel{+}{=} K(S_1) + K(S_2|p^*_{S_1})$.

The conditional in the second term of the last equation indicates that the identification of the square by $p^*_{S_1}$ can be reused for describing S_2. If they are nevertheless of equal complexity as S, then signifies that S_1 and S_2 contain no additional information that is not already required to describe S. *K-decomposability* thus refers to the possibility of decomposing a set into multiple sets without increasing the descriptive complexity. The description of the total set can be decomposed into a separate description of subsets. Note that this definition has the same form as the additivity rule (Equation (2) in section 3.3), which always holds for Kolmogorov complexities. In the case of K-decomposability, however, we only get additivity if the set decomposition does not bring in new 'elements' on the right side of the equation that are not present on the left side.

Applied to the example of *something stimulating* we can see that it is in fact *K-decomposable*. The three variables *red*, *food*, and *tickle* are identified with three separate functions and so $K(S) \stackrel{+}{=} K(red) + K(food|p^*_{red}) + K(peck|p^*_{red}, p^*_{food})$. The Kolmogorov complexity of the set S is the sum of the KC of the three subsets. On the other hand, a concept that is appropriately uniform cannot be decomposed. Consider the set of all squares. To decompose this set, we would have to segregate the squares according to a certain criterion. For example, we could apply a threshold on their size to distinguish small from large squares. But then we have to include this criterion in

the indicator functions of both subsets, which makes the total description larger than the original one and invalidates the decomposition. Thus, we can plausibly say that 'square' is not *K-decomposable*.

Using the notion of *K-decomposable* sets we can then define when a variable V is uniform, in accordance with the informal characterization above:

Definition 5. *V is a **uniform** variable if the set S_V that corresponds to V is not K-decomposable*

In other words, the variables that we are looking for are those that attach to a unified characteristic, such as *red* or *brightness*, guaranteed by the fact that the concepts used are not *K-decomposable*. Importantly, for any explanation and element x there is a wide range of uniform variables that can be used. A specific x could be both red (one uniform variable), and a rectangle (a second uniform variable) and a large object (a third uniform variable) at the same time. Furthermore, variables at various levels of abstraction can be uniform: *scarlet* is uniform, as is *red* and *colour*. Which uniform variables one chooses then depends on the specific explanation (and to be more precise the domain of that explanation), to which we turn next.

4.3 Formal definition of optimal variable in an explanation

To arrive at our final definition of optimal variables for an explanation we then also need the notion of *K-composability*. The idea here is that in explanations we often use more than one variable to capture the set of inputs X for whose outputs Y the explanation is supposed to provide additional insight. The interaction of these different variables needs to be accounted for, as they should be complementary. To capture this aspect we therefore define *K-composability* as follows:

Definition 6. *A set S is K-composable if there exist different and non-empty subsets S_1 and S_2 such that:*

- $S_1 \setminus S_2 \neq \emptyset$,
- $S_2 \setminus S_1 \neq \emptyset$,
- $S = S_1 \cap S_2$, *and*
- $K(S) \stackrel{+}{=} K(S_1) + K(S_2|p^*_{S_1})$.

Using both K-decomposability (based on the union of sets) and K-composability (based on the intersection of sets) we can then define when variables are of the optimal degree of abstraction for an explanation of an input x. Here, we will say that an explanation $G(x)$ of $b(x)$ regarding element x aims to cover as large a set as possible while still using only patterns (of dependence) relevant to x.

Definition 7. *An optimal explanation G of x has domain $S_G \subseteq S_b$, where S_G is a maximal non-K-decomposable set which contains x*

For optimal explanations of x there is then a guarantee that S_G contains as many inputs as possible, while it does not include irrelevant information for x (as in this case it would be possible to K-decompose S_G).

Our definition then states that the optimal variables together identify this subset S_G of inputs in a complementary fashion, building on the definition of a uniform variable.

Definition 8. *Uniform variables $V_1, \ldots V_n$, associated with set $S_{V_1}, \ldots S_{V_n}$, are **optimal variables in explanation** $G(x)$ of input x for $b(x)$ if the set S_G associated with the explanation is such that the sets $S_{V_1}, \ldots S_{V_n}$ corresponding to uniform $V_1, \ldots V_n$ K-compose S_G.*

Figure 2 helps to visualize what this definition states. Our choice of S_G as the maximal subset of S_b that is non-K-decomposable and contains x is illustrated on the left. On the right we see how an explanation of x is then built up using optimal variables. The composability requirement states that if we use different variables in an explanation then they have to overlap, to together characterize S_G. However, they have to do so in complementary fashion (and with a minimal number of variables). So, these sets will be similar to those seen on the right-hand side in the image. For example, if S_G is the set of all big, red rectangles then it is K-composed of the sets S_{V1}: 'rectangles', S_{V2}: 'big objects' and S_{V3}: 'red objects'. The requirement that the difference sets are non-empty helps to exclude the possibility to further compose the set of rectangles based on the set of 'quadrilaterals' or 'polygons'. This is not a valid composition of 'rectangles' by our definition since 'rectangles' minus 'quadrilaterals' is the empty set. This, together with the requirement that the Kolmogorov complexity does not increase through K-composition, helps prevent the move to more abstract concepts.

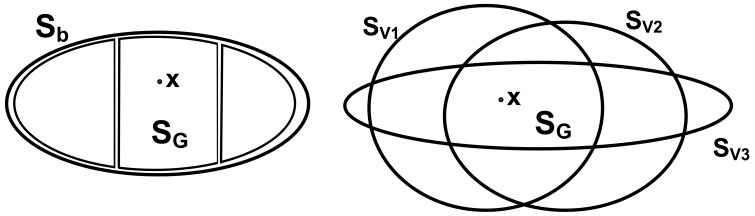

Fig. 2. An explanation G explains inputs in subset S_G of the positive subset S_b (left), and does so using the intersection of global variables S_{V1}, S_{V2} and S_{V3} (right).

5 Application of the definition to the pigeon case

If we apply this proposed definition to the examples depicted in Section 4.1, we see that it tracks exactly the judgements we are inclined to make. According to our definition, we should prefer *red* over *scarlet* because the resulting explanation covers a larger subset (namely all red stimuli rather than only the scarlet stimuli), while *red* can be defined in simpler (i.e. shorter) terms than as a disjunction of the different shades of red. The subset S_G is in this case the set of all red stimuli, assuming that x is a specific red stimulus. While we could describe this with variables of different shades of red (which are uniform variables), this increases the Kolmogorov complexity of the set as red is not *K-decomposable*. Hence, we should prefer *red* over a disjunction of shades of red. We should also prefer it over *scarlet*, as *scarlet* does not K-compose S_G on its own.

Finally, more abstract variables such as *colour* are ruled out as composition of more abstract variables is more complex than simply using *red* (violating the last condition of K-composition). However, had the bird pecked only at scarlet stimuli, our definition would state that *scarlet* is the optimal variable to use. In that case, S_b would not have contained other shades of red and so likewise $S_G \subseteq S_b$ would have been restricted to the specific set of scarlet stimuli, which are then captured by the uniform variable *scarlet*.

Furthermore, we should prefer *red* over *something stimulating* in the situation where the bird pecks at a wider range of stimuli. Despite the broader reach of *something stimulating* it is *K-decomposable* in terms of *red*, *food* and *tickle*. As a result, we first fix S_G as one of the maximal non-K-decomposable subsets, in the case of a red stimulus this will be *red*. This means that *something stimulating*, the variable that covers all S_b, is ruled as being too abstract. Instead, we should look at the minimal number of non-K-decomposable sets that together K-compose the smaller set S_G, which is simply *red* again. Should we want a more general explanation of the behaviour of the pigeon then we can simply go for the disjunction of the explanations corresponding to our K-decomposed subsets: *red* $\vee$ *food* $\vee$ *tickle*. For the follow-up examples we get again the desired results: *red* is preferable to *redow* because *redow* is *K-decomposable* in terms of *red* and *yellow* (which affects the choice of S_G), whereas *bright coloured* is preferable because it is not *K-decomposable*, again assuming that the set of positive instances that we aim to explain is in the first case that of red and yellow stimuli and in the second case that of bright coloured stimuli.

To consider this in the pigeon example just discussed, we can imagine a setting in which the brain of the pigeon is studied and neural signals are measured. Based on these measurements it is observed that when a particular part of the brain gets stimulated it cause the pecking. In such a context, with the knowledge of what's going on in the brain, 'something stimulating' can provide a good (uniform) explanation for the pecking. For each of the different stimuli (a bright color, food or tickle), a similar process in the brain can be observed. This then changes what the optimal variable is, as it shows that *something stimulating* is in fact not *K-decomposable* as we initially thought.

6 Explanations in AI

A popular option in XAI for explaining black-box algorithms is to fit decision trees to them, which aim to approximate the input-output relation of b [11, 16]. They offer a human-understandable description, thanks to the explicit variables and clear decision paths (for trees that are not too large). Can they represent/describe the level of abstraction we defined in this paper? Decision trees are based on clauses that form conditions on the input variables by conjunctions, negations, and disjunctions. Also rule-based systems are based on such clauses.

Figure 3 shows the decision tree for the Pigeon2 example in which the color might be red or yellow. The decision tree provides an explanation for all positive instances, where each node is an optimal variable for an explanation and each leaf represents the right level of abstraction for an explanation of an individual outcome (an S_G-set).

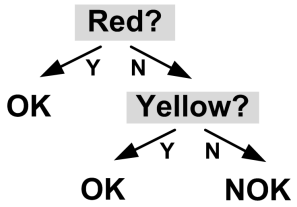

Fig. 3. Decision tree for the Pigeon2 example (Red or Yellow).

In this example, the decision tree is the shortest description of the partitioning. But this is not always true: for more regular structures, a shorter description is possible. Consider the partitioning of a chessboard into 64 squares. Describing all black squares separately results in a large tree. An algorithm can do this in a more succinct way by exploiting the regularities of a chessboard. The KC of the black squares is smaller than a literal enumeration of all squares.

Likewise, the set of images containing a pattern, such as a rectangle which is recognized by a NN, cannot be explained succinctly by a decision tree. The NN employs multiple layers of various operations applied to the image pixels to achieve the recognition. This cannot be described by simple clauses. Patterns are the regularities that reduce the KC, while constraints on parameters do not reduce the KC. Consider the set S_b of the rectangles of a certain size and color. An explanation for S_b is K-composed of 3 optimal variables: the rectangular shape, the size and the color. The variables can be extracted without increasing the KC. The first one describes the pattern. The second and third are constraints. Such constraints can be formed by conditions on the input variables, but also conditions on the parameters of the patterns. Conditions can be described succinctly by a decision tree or rules. Patterns, however, cannot. To overcome this challenge, [14] propose to use decision trees containing *prototypes* that are representative for a set of similar instances. By checking against the prototype in the node it is possible to classify a case using more abstract concepts.

7 Conclusion

How abstract should variables in explanations be? We have proposed an account based on Kolmogorov complexity which, although not computable, gives us a formal definition of the optimal degree of abstraction. As shown in section 5, our formal definition handles the examples in the philosophical literature well. We have done so by first defining the notion of a uniform, i.e. undecomposable, variable. Which of these uniform variables is optimal for a given explanation is then based on what variables can be combined to characterize the patterns of dependence captured by the explanation with minimal Kolmogorov complexity. Ultimately, therefore, we approach the problem of abstraction by arguing that the optimal degree of abstraction is that which leads to the least complex description of the patterns and constraints in the explanation. As abstraction is precisely meant to simplify description, we consider it a natural link to say

that the optimal degree of abstraction is that which optimally reduces the complexity of descriptions.

References

1. Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al.: Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information fusion **58**, 82–115 (2020)
2. Balayn, A., Soilis, P., Lofi, C., Yang, J., Bozzon, A.: What do you mean? interpreting image classification with crowdsourced concept extraction and analysis. In: Proceedings of the Web Conference 2021. pp. 1937–1948 (2021)
3. Beisbart, C., Räz, T.: Philosophy of science at sea: Clarifying the interpretability of machine learning. Philosophy Compass **17**(6) (2022)
4. Biswas, S., Corti, L., Buijsman, S., Yang, J.: Chime: Causal human-in-the-loop model explanations. In: Proceedings of the AAAI Conference on Human Computation and Crowdsourcing. vol. 10, pp. 27–39 (2022)
5. Blanchard, T.: Explanatory abstraction and the goldilocks problem: Interventionism gets things just right. The British Journal for the Philosophy of Science (2020)
6. Buijsman, S.: Defining explanation and explanatory depth in XAI. Minds and Machines **32**(3), 563–584 (2022)
7. Franklin-Hall, L.R.: High-level explanation and the interventionist's 'variables problem'. The British Journal for the Philosophy of Science (2016)
8. Ghorbani, A., Wexler, J., Zou, J.Y., Kim, B.: Towards automatic concept-based explanations. Advances in Neural Information Processing Systems **32** (2019)
9. Gilpin, L.H., Paley, A.R., Alam, M.A., Spurlock, S., Hammond, K.J.: "Explanation" is not a technical term: The problem of ambiguity in XAI. arXiv preprint:2207.00007 (2022)
10. Grünwald, P.: The minimum description length principle. MIT Press, Cambridge, MA (2007)
11. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM computing surveys **51**(5) (2018)
12. Li, M., Vitányi, P.: An Introduction to Kolmogorov Complexity and its Applications. Springer (1997)
13. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. Advances in neural information processing systems **30** (2017)
14. Nauta, M., Van Bree, R., Seifert, C.: Neural prototype trees for interpretable fine-grained image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14933–14943 (2021)
15. Ribeiro, M.T., Singh, S., Guestrin, C.: "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. pp. 1135–1144 (2016)
16. Sagi, O., Rokach, L.: Explainable decision forest: Transforming a decision forest into an interpretable tree. Information Fusion **61**, 124–138 (2020)
17. Weatherson, B.: Explanation, idealisation and the goldilocks problem. Philosophy and Phenomenological Research **84**(2), 461–473 (2012)
18. Woodward, J.: Making things happen: A theory of causal explanation. Oxford university press (2005)
19. Woodward, J.: Explanatory autonomy: the role of proportionality, stability, and conditional irrelevance. Synthese **198**, 237–265 (2021)
20. Yeh, C.K., Kim, B., Arik, S., Li, C.L., Pfister, T., Ravikumar, P.: On completeness-aware concept-based explanations in deep neural networks. Advances in Neural Information Processing Systems **33**, 20554–20565 (2020)