PAC Learning and Occam's Razor: Probably Approximately Incorrect

Daniel A. Herrmann*†

Computer scientists have provided a distinct justification of Occam's Razor. Using the probably approximately correct framework, they provide a theorem that they claim demonstrates that we should favor simpler hypotheses. The argument relies on a philosophical interpretation of the theorem. I argue that the standard interpretation of the result in the literature is misguided and that a better reading does not, in fact, support Occam's Razor at all. To this end, I state and prove a very similar theorem that, if interpreted the same way, would justify the contradictory Anti-Occam's Razor—the principle that we should favor more complex hypotheses.

1. Introduction. Occam's Razor roughly states that "all other things being equal, the simplest hypothesis is the best hypothesis." Although originally invoked by William of Ockham to shave away what he took to be the excessive metaphysical commitments of the scholastics, scientists and philosophers alike appeal to it today. It arises in many scientists' commitment to elegance and simplicity in their theories and in discussions of parsimony in the philosophy of science. Although Occam's Razor is quite intuitive—it can be argued we use it every day of our lives—a question remains: Why should we adhere to such a principle?

Broadly speaking, there are three main interpretations of the razor: metaphysical, pragmatic, and epistemic. The metaphysical interpretation is that Occam's Razor states a property about the world: we should expect the world to be simple. The pragmatic interpretation states that using simple

Received September 2018; revised October 2019.

*To contact the author, please write to: University of California, Irvine, Irvine, CA 92697; e-mail: daherrma@uci.edu.

†I want to thank Jan-Willem Romeijn and Tom F. Sterkenburg for invaluable early discussions and feedback. I also want to thank Simon Huttegger, Brian Skyrms, Darcy Otto, Aydin Mohseni, Gerard Rothfus, Saira Khan, Bruce Rushing, Tom Colclough, and two anonymous referees for helpful comments on earlier drafts.

Philosophy of Science, 87 (October 2020) pp. 685–703. 0031-8248/2020/8704-0007$10.00
Copyright 2020 by the Philosophy of Science Association. All rights reserved.

https://doi.org/10.1086/709786 Published online by Cambridge University Press

hypotheses makes our practical inference easier: it is easier for us to work with simple hypotheses. The epistemic interpretation states that Occam's Razor is rather best understood as a principle of methodology: favoring simpler hypotheses over more complex ones helps us reach the truth in our inquiry.

Occam's Razor is discussed in the philosophy literature. In particular, Sober (2015) traces Occam's Razor in the history of philosophy and science, drawing from philosophers such as Aristotle, Descartes, Hume, and Kant. He follows the razor to its modern day incarnations in probability theory, surveying Bayesian and frequentist justifications and uses of simplicity in inquiry.

There are also justifications in the computational learning theory and machine learning literature. In this article I aim to clarify and argue against one widely used justification of the epistemic value of Occam's Razor—a theorem from probably approximately correct (PAC) learning. This result, first proved by Blumer et al. (1987), is still frequently reproduced and cited. In their words, they show that "under very general assumptions, Occam's Razor produces hypotheses that with high probability will be predictive of future observations" (378). Even Valiant, the computer scientist who first introduced the PAC framework, references this result in his recent book (Valiant 2013).

The argument from PAC learning is just one of many seeking to justify the epistemic value of Occam's Razor. There are arguments from Kolmogorov complexity and computability theory in the form of Solomonoff induction (see Ming and Vitányi 1990) and from attempts to minimize changes of opinion to increase efficiency while still satisfying the strongest convergence to the truth available (see Kelly 2004). Thus, this article is not meant to be a takedown argument of every justification of the epistemic virtue of Occam's Razor, but rather it is meant to clarify what exactly the result from PAC learning does and does not show.

It is also worthwhile to consider the philosophical implications of getting a better grip on arguments purported to justify Occam's Razor. Consider the argument from Solomonoff induction mentioned above. Although on the surface it seems plausible, Sterkenburg (2016) recently showed that using Solomonoff induction to justify Occam's Razor does not work. Although we can view such a result as negative, such projects also have positive implications. By demonstrating that simplicity played no role in the supposed success of Solomonoff induction, Sterkenburg was able to reveal where Solomonoff induction was philosophically interesting: its inductive assumption of an effective environment. By getting clarity on the structure of such arguments, we can uncover where the real philosophical interest lies.

In section 2 I introduce the PAC learning framework. In section 3 I introduce the theorem given to justify Occam's Razor in the PAC learning literature. A proof is provided in appendix A. I then motivate this article by

https://doi.org/10.1086/709786 Published online by Cambridge University Press

stating a similar theorem that, by the same reasoning, would justify Anti-Occam's Razor: the principle that we should favor complex hypotheses over simple ones. A proof of this theorem is in appendix B. This is meant to demonstrate the misinterpretation of the Occam result. In section 4 I discuss what both these theorems actually say and why they do not conflict, and I suggest a different interpretation than the one standard in the literature. Rather than providing a justification for Occam's Razor, what we find is that we can learn when we are able to restrict our full hypothesis space to a properly growing subspace and we are guaranteed to output an approximately correct hypothesis. This requires that each hypothesis space in the family exhibits a certain similarity property. It is this requirement, I argue, that is the unstated assumption that is doing most of the work. In section 5 I give two further arguments meant to cast doubt on this justification of Occam's Razor. In section 6 I offer some brief concluding remarks.

2. PAC Learning Framework. One of the most studied and discussed frameworks in computer science and machine learning is PAC learning. Drawing on ideas from computational complexity and function approximation, Valiant introduced the PAC learning framework, to show that it is possible to design machines that fulfill the following three desiderata:

> 1. The machines can provably learn whole classes of concepts. Furthermore, these classes can be characterized.
> 2. The classes of concepts are appropriate and nontrivial for general-purpose knowledge.
> 3. The computational process by which the machines deduce the desired programs requires a feasible (i.e., polynomial) number of steps.
>
> (1984, 1134)

Although framed here in terms of machines, I follow later authors and discuss learning methods, or learning algorithms. Informally defined, a learning method is a function from data to a concept.[1]

PAC learning, then, is a model used to understand which classes of concepts learning algorithms can learn in a feasible number of steps. Informally, in this context we say an algorithm can "learn" if it has a high chance of producing a predictively accurate concept. "Feasible" in this context is cashed out as polynomial, both for the time complexity of the learning algorithm and the number of examples needed to learn the function. We can think of time and examples as resources that the algorithm requires to learn. When we say that an algorithm requires an amount of some resource such as time

1. I provide a formal definition of both a learning method and a concept shortly.

https://doi.org/10.1086/709786 Published online by Cambridge University Press

or sample size "polynomial in" some other quantity X, this means that, as X grows, the amount of resources needed by the algorithm to learn can be upperbounded by a polynomial function of the quantity of X.[2]

In what follows I present the framework following the modern way PAC learning is described and used in the Occam proof (I follow Blumer et al. 1987 in particular). Let the hypothesis class $\mathcal{H}_n$ be a set of functions that each map from a domain $\mathcal{X}$ to a range $\mathcal{X}'$. There is some $f \in \mathcal{H}_n$ that is the true function, or concept, that the learner is trying to learn. An observation is a pair $(x_i, f(x_i))$ where $x_i \in \mathcal{X}$. Observations of f are made independently according to a probability distribution P on $\mathcal{X}$ that does not change over time. A *sample* of f is a sequence of m observations $(x_1, f(x_1))$, $(x_2, f(x_2))$, . . . , $(x_m, f(x_m))$.

For example, $\mathcal{X}$ could be the set of all strings of 0's and 1's of length n, which we denote as $\{0, 1\}^n$, and $\mathcal{H}_n \subseteq \mathcal{F}_n$, where $\mathcal{F}_n$ is the set of all n-place Boolean functions. I follow the custom of using this as the canonical example. In particular, even in cases in which $\mathcal{X}$ is not a set of finite binary strings, the proofs of Occam's Razor always assume some standard encoding of all $x \in \mathcal{X}$ and all $x' \in \mathcal{X}'$ such that this setup is equivalent to these cases (see Blumer et al. 1987).

A learning algorithm is an algorithm that maps each sample of any function $f \in \mathcal{H}_n$ to a function $h \in \mathcal{H}_n$ that is consistent with the sample (i.e., for all x_i in the sample, $h(x_i) = f(x_i)$) in time polynomial in the length of the given sample.

For an example of the kind of learning context described by the framework, consider the learning problem in which we are trying to identify pictures of cats. Our strings could encode the pixels in an image (e.g., a 1,024 length string could be a string representing a 32×32 image of black and white pixels). Our labels are 1 corresponding to cat and 0 corresponding to noncat. For each set of n-length strings, our hypothesis space is some subset of all 2^{2^n} Boolean functions. This subset has the true cat function in it for images of a certain size, but we do not know which one it is. Here is a sketch of a general learning procedure: any time the learner observes a picture with a label, it deletes any hypothesis in the hypothesis space that assigns that picture the wrong label. If the image is of a cat, but a hypothesis says 0 (noncat), the learner deletes that hypothesis. When queried for an affirmative hypothesis, the learner chooses one of the remaining hypotheses according to some decision method.

Once we specify our error tolerances, we can use the tools of PAC learning to calculate how many examples we would need to see to learn to distinguish cats from noncats for images encoded by strings of different length.

2. For an excellent introduction to computational complexity and its interest to philosophers, see Aaronson (2013).

https://doi.org/10.1086/709786 Published online by Cambridge University Press

This is where the sequence of hypothesis spaces $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_n$ is playing a role in the formal framework. When considering images encoded with a string of length n, there is some hypothesis space $\mathcal{H}_n$ containing all the possible concepts for images of that size, which is a subset of all n-place Boolean functions. Intuitively, increasing the size of our pictures means that we have to consider more hypotheses. If the number of examples needed by the algorithm to learn grows polynomially with the size of our image, then the family of hypothesis spaces is PAC learnable.

We can now give a formal definition of learnability within the framework.

> **Definition 1.** A family of hypothesis spaces $\mathcal{H} = \{\mathcal{H}_1, \mathcal{H}_2, \ldots\}$ is polynomially (or PAC) learnable if there exists a learning algorithm A and a function $m(\varepsilon, \delta, n) : (0, 1) \times (0, 1) \times \mathbb{N} \to \mathbb{N}$ polynomial in the arguments $1/\varepsilon$, $1/\delta$, n such that, for every $n > 0$, every concept $f \in \mathcal{H}_n$, every probability distribution P on $\{0, 1\}^n$, every $\varepsilon \in (0, 1)$, and every $\delta \in (0, 1)$, A on input of a sample of any size greater than or equal to $m(\varepsilon, \delta, n)$ (chosen according to P) with probability $1 - \delta$ outputs a hypothesis $h \in \mathcal{H}_n$ that disagrees with the true function f on an $x \in \mathcal{X}$ drawn randomly according to P with at most probability ε.

This definition is at the heart of PAC learning, whence it derives its name—probably approximately correct learning. Intuitively, a family of hypothesis spaces is PAC learnable if we have an algorithm that requires only a reasonable (polynomial) number of examples as n increases to output a hypothesis that is within the ε-neighborhood of f with probability $1 - \delta$. Since $\mathcal{H}_n \subseteq \mathcal{F}_n$, we can also think of n increasing as the length of the binary strings the algorithm needs to learn to classify increasing. A hypothesis h is within the ε-neighborhood of f if the probability that $h(x) \neq f(x)$ when x is drawn randomly according to P is at most ε. Informally, δ corresponds to "probably" and ε corresponds to "approximately"—that is, δ represents how confident you want to be that the algorithm outputs a function h close to f, and ε specifies how much disagreement between h and f you are willing to tolerate.

Thus, by applying the definition to a given case, we are able to see whether a family of hypothesis spaces is learnable. We have the tools we need to understand the theorems in the following section, as well as the proofs of the theorems in appendixes A and B.

3. Occam and Anti-Occam

3.1. Occam's Razor. Although there are subtly different ways to try to prove Occam's Razor in the PAC learning framework (e.g., Wolf 1997; Ming, Tromp, and Vitányi 2003), they all follow a pattern similar to the

https://doi.org/10.1086/709786 Published online by Cambridge University Press

original proof given by Blumer et al. For the purposes of this article, the proof I include follows closely the presentation of the original Blumer et al. paper (1987). I follow parallel reasoning in my proof of Anti-Occam's Razor given in appendix B.

In order to sharpen the notion of description length, I suppose we have fixed some standard binary encoding of the hypotheses in the family $\mathcal{H}$, in the sense that we have an encoding function $\xi : \cup_n \mathcal{H}_n \rightarrow \mathcal{L}$, where $\mathcal{L}$ is the set of possible function names in our encoding. I assume that the alphabet of $\mathcal{L}$ is $\{0, 1\}$ so that $\mathcal{L}$ is some binary encoding. Thus, the name of f is $\xi(f)$, and the length of $\xi(f)$ is $\lambda(\xi(f))$, which I often abbreviate as $\ell(f)$. It is important to note that, of course, $\ell(f)$ is relative to the choice of ξ. The complexity of f is then $\ell(f)$. This is meant to match our intuition that more complex hypotheses require a longer description.

Now that we have a precise notion of description length, we are ready to define an Occam algorithm.

> **Definition 2.** A learning algorithm $\mathcal{O}$ is an Occam algorithm if for every $n > 0$ with respect to the hypothesis space $\mathcal{H}_n$, on input of a sample of any $f \in \mathcal{H}_n$ of (sufficiently large) size m, $\mathcal{O}$ produces a hypothesis h such that $\ell(h) \leq p(n)m^{\alpha}$, where p is a polynomial and $0 \leq \alpha < 1$.[3]

This definition says that an Occam algorithm is guaranteed to output a hypothesis with a short name. This allows us to restrict the hypothesis space from the full $\mathcal{H}_n$ to include only hypotheses h of length $\ell(h) \leq p(n)m^{\alpha}$. Note that since the names are binary strings, the cardinality of the reduced hypothesis space is at most $2^{p(n)m^{\alpha}}$. We are now in a position to state the main theorem.

> **Theorem 1.** For a hypothesis class $\mathcal{H}$, for each Occam algorithm $\mathcal{O}$ there is a polynomial function $m(\varepsilon, \delta, n)$ such that the following holds. For every $n > 0$, every concept $f \in \mathcal{H}_n$, every probability distribution P on $\{0, 1\}^n$, every $\varepsilon \in (0, 1)$, and every $\delta \in (0, 1)$, on input of a sample of any size greater than or equal to $m(\varepsilon, \delta, n)$ chosen according to P, $\mathcal{O}$ outputs a hypothesis $h \in \mathcal{H}_n$ that with probability $1 - \delta$ disagrees with the true function f on an $x \in \mathcal{X}$ drawn randomly according to P with at most probability ε.[4]

This theorem is presented as an epistemic justification of Occam's Razor. As Blumer et al. write:

> Given two explanations of the data, all other things being equal, the simpler explanation is preferable. This principle is very much alive today in

3. Definition 2 closely follows definition 2.2 in Blumer et al. (1987). Minor changes have been made in order to make the notation consistent with that used throughout this article.

4. This follows theorem 2.3 in Blumer et al. (1987). As above, minor changes have been made in order to make the notation consistent with that used throughout this article.

https://doi.org/10.1086/709786 Published online by Cambridge University Press

> the emerging science of machine learning, whose expressed goal is often to discover the simplest hypothesis that is consistent with the sample data. As laudable as this goal may seem, whether in the area of machine learning or in science as a whole, one can still ask why the simplest hypothesis for a given sequence of observations should perform well on further observations taken from the same source. After all, the real value of a scientific explanation lies not in its ability to explain past events, but in predicting events that have yet to occur. We show that, under very general assumptions, Occam's Razor produces hypotheses that with high probability will be predictive of future observations. (1987, 377–78)

It is clear from the language in this passage that Blumer et al. are concerned with the assumption that Occam's Razor is a good epistemic principle. They are not suggesting that scientists and computer scientists treat it as a pragmatic principle but rather that they actually expect it to help us choose good hypotheses for making predictions. Further, they view their proof of theorem 1 as providing a justification of this principle.

In order to highlight that philosophical interpretation that some computer scientists give this theorem, consider also what Kearns and Vazirani write:

> Despite its long and illustrious history in the philosophy of science and its extreme generality, there is something unsatisfying about the notion of an Occam algorithm. After all, the primary goal of science (or more generally, of the learning process) is to formulate theories that accurately predict future observations, not just to succinctly represent past observations. In this chapter, we will prove that when restricted to the probabilistic setting of the PAC model, Occam algorithms do indeed have predictive power. This provides a formal justification of the Occam principle, albeit in a restricted setting. (1994, 32)

We see both that they are interested in providing a formal justification for a philosophical principle—indeed, the authors call their take on the theorem a "philosophical interpretation" (32)—and that they take theorem 1 to be such a justification.

We have before us the canonical PAC-learning theorem demonstrating the epistemic virtue of simplicity. We turn to a puzzle—a similar theorem demonstrates the virtue of complexity.

3.2. Anti-Occam's Razor. If one takes theorem 1 to justify Occam's Razor, one might take the theorem in this section to justify the principle that I call Anti-Occam's Razor—the principle that we should favor complex hypotheses. I first define an Anti-Occam algorithm.

https://doi.org/10.1086/709786 Published online by Cambridge University Press

Definition 3. A learning algorithm $\mathcal{A}$ is an Anti-Occam algorithm if for every $n > 0$ with respect to the hypothesis space $\mathcal{H}_n$, on input of a sample of any $f \in \mathcal{H}_n$ of (sufficiently large) size m, $\mathcal{O}$ only outputs one of the top $\mathcal{C}$ most complex hypotheses, where $\mathcal{C} = 2^{p(n)m^{\alpha}}$, $p(n)$ is a polynomial, and $0 \leq \alpha < 1$.

This definition says that an Anti-Occam algorithm outputs a hypothesis only if it is among the most complex. We are now in a position to state the main theorem of this section. Although the proof is almost identical to the proof of theorem 1, I provide it in appendix B in full so that the reader may compare the two and get an understanding of what is doing the work, so to speak.

Theorem 2. For a hypothesis class $\mathcal{H}$, for each Anti-Occam algorithm $\mathcal{A}$ there is a polynomial function $m(\varepsilon, \delta, n)$ such that what follows holds. For every $n > 0$, every concept $f \in \mathcal{H}_n$, every probability distribution P on $\{0, 1\}^n$, every $\varepsilon \in (0, 1)$, and every $\delta \in (0, 1)$, on input of a sample of any size greater than or equal to $m(\varepsilon, \delta, n)$ chosen according to P, $\mathcal{A}$ outputs a hypothesis $h \in \mathcal{H}_n$ that with probability $1 - \delta$ disagrees with the true function f on an $x \in \mathcal{X}$ drawn randomly according to P with at most probability ε.

If the proof of theorem 1 in appendix A is a proof of Occam's Razor, then the proof of theorem 2 in appendix B is a proof of Anti-Occam's Razor. What are we to make of this? It cannot be the case that we have just proved two contradictory epistemic principles.

4. Interpretation. The standard interpretation of theorem 1 is meant to support the use of Occam's Razor in our inquiry, at the very least, in the context of PAC learning.[5] However, theorem 2 confuses this. In order to understand how we should interpret these results, I examine how the term "exists" is used in the definition of PAC learning (definition 1). The result is that the way existence is used in the PAC learning definition makes the correct interpretation of the theorems far from a justification of either razor.

4.1. Existence in the Definition of PAC Learnability. Informally, the definition of PAC learnability states that a family of hypothesis spaces is learnable if there exists a learning algorithm that only needs a polynomial number of examples in order to find a PAC hypothesis. This is reasonable—the polynomial bound is meant to capture the notion of a feasible number of steps (Valiant 1984), and if no such algorithm exists this means that there is no method that can feasibly learn the family.

5. Although we saw above how the authors have an eye toward justifying its use in science as well.

https://doi.org/10.1086/709786 Published online by Cambridge University Press

What would it mean for there not to be an algorithm that can do this? This is important to clarify, as definition 1, theorem 1, and theorem 2 all make use of a condition of the form, "if there exists such and such an algorithm." For example, in the case of definition 1, what stops me from saying that every hypothesis space is learnable? In what case does no such algorithm exist?

To understand this, we need to look closely at definition 1. The definition states that there must exist one "learning algorithm A and a function $m(\varepsilon, \delta, n) : (0, 1) \times (0, 1) \times \mathbb{N} \rightarrow \mathbb{N}$ polynomial in the arguments $1/\varepsilon$, $1/\delta$, n such that, for every $n > 0$, every concept $f \in \mathcal{H}_n$, every probability distribution P on $\{0, 1\}^n$, every $\varepsilon \in (0, 1)$, and every $\delta \in (0, 1)$, A on input of a sample of any size greater than or equal to $m(\varepsilon, \delta, n)$ (chosen according to P) with probability $1 - \delta$ outputs a hypothesis." The algorithm has to work for any choice of δ and ε, and for every concept $f \in \mathcal{H}_n$. This is important. This means that I cannot simply say a good algorithm exists for $\mathcal{H}$ if you give me an algorithm that correctly identifies a hypothesis g in the ε-neighborhood of the true function f by luck. Clearly, if this were allowed, then every $\mathcal{H}$ would be learnable, as there is some algorithm whose first guess at every n simply is the true hypothesis for some particular choice of $f \in \mathcal{H}_n$. The same algorithm has to work regardless of which $f \in \mathcal{H}_n$ is the true function for every $n > 0$.

Consider our earlier cat example. Although the command "learn the cat function" sounds like a good description of the problem, in the PAC framework it would be more correct to say "learn which of the functions in the hypothesis space is the true function." We do not want our algorithm to work only for the actual cat function because we do not know beforehand which function in the space is the true one. This is why the definition says that a "family of hypothesis spaces" is learnable; it does not say that a "function" is learnable.

The intuitive way to think of this is that when we ask whether a certain problem is PAC learnable, we are not asking whether a particular function in a hypothesis class is learnable but whether the class is learnable. That is, the algorithm should work regardless of which function in the hypothesis space turns out to be the true one. We have seen, then, that in the definition of PAC learning, a necessary condition for a good algorithm existing for a certain family $\mathcal{H}$ is that, for each $n > 0$, the algorithm must work regardless of which $f \in \mathcal{H}_n$ is the true hypothesis.

4.2. Implications for the Occam Theorem. Informally, theorem 1 states that every Occam algorithm for a family of hypothesis spaces is a good learner, in the PAC learning sense. That is, it needs only a feasible number of examples. However, in general, we are not guaranteed that any Occam algorithms exist for a specific family of hypothesis spaces. Thus, we want

https://doi.org/10.1086/709786 Published online by Cambridge University Press

to know the conditions under which there exists an Occam algorithm for a family of hypothesis spaces.

What would it mean for there not to be an Occam algorithm for a family of hypothesis spaces? A learning algorithm $\mathcal{O}$ is Occam if the length of the name of any function it outputs is bounded by a term polynomial in n. Thus, for each $n > 0$, the Occam algorithm restricts the hypotheses under consideration from the full $\mathcal{H}_n$ given by the family to a subset of $\mathcal{H}_n$, whose cardinality is $2^{p(n)\cdot m^{\alpha}}$. For ease of reference, let this set be called $\mathcal{H}_n^{\mathcal{O}}$. Then, clearly, $\mathcal{H}_n^{\mathcal{O}} \subseteq \mathcal{H}_n$.

Now it is clear what is driving the result. Informally, if an algorithm is able to restrict its attention to a subset of the actual hypothesis space, then it is able to learn.

This raises a problem that is not present in the standard PAC learning context. Usually, since our algorithm considers the full hypothesis space, we are guaranteed that there is at least one function in the ε-neighborhood of the true function that will always be consistent with any sample: the true function itself. Thus, even though learning algorithms only output a hypothesis if it is consistent with all of the examples, we are guaranteed that we will always have such a hypothesis to output, because the true hypothesis is in the set of hypotheses under consideration.[6]

This is not necessarily so for Occam algorithms. Because such algorithms only consider a subset of the hypothesis space, $\mathcal{H}_n^{\mathcal{O}}$, it is entirely possible that the true function lies outside of this subset. One might think that this is not necessarily a problem because all that needs to lie within $\mathcal{H}_n^{\mathcal{O}}$ is a hypothesis within the ε-neighborhood of the true function. However, even this would not be entirely satisfying, for, since such a hypothesis would not be identical to the true hypothesis, it would disagree on some examples, and it is possible that it would be eliminated during training, leaving $\mathcal{O}$ no hypothesis to output. That is, even if there is some $g \in \mathcal{H}_n^{\mathcal{O}}$ that is in the ε-neighborhood of the true f, we have no guarantee that it will not get eliminated during training. Again, this is very different from the more standard case when the true f is guaranteed to be in the set of hypotheses, in which we never need to worry that we will not output a hypothesis.

Furthermore, it is important to note that we also have no guarantee that there will be some $g \in \mathcal{H}_n^{\mathcal{O}}$ that is in the ε-neighborhood of the true f. It is entirely possible that in choosing our subset we only end up with hypotheses that lie outside of the ε-neighborhood of the true f. During training, these will all be eliminated with probability $1 - \delta$, leaving $\mathcal{O}$ with no hypothesis

6. One might be concerned here about error in the observations—could this not rule out the true hypothesis? This is a genuine concern. However, in the PAC learning framework, samples are assumed to be without error.

https://doi.org/10.1086/709786 Published online by Cambridge University Press

to output. Furthermore, even if by chance one is not eliminated, this will be a hypothesis outside of the ε-neighborhood of the true function.

In the standard PAC learning context, where we consider all hypotheses, the algorithm always outputs a hypothesis because the true f can never be eliminated. A hypothesis family is PAC learnable if the hypothesis it outputs is guaranteed to be within the ε-neighborhood of the true f.

When using an Occam algorithm, however, we might end up in a situation in which there is no hypothesis that $\mathcal{O}$ can output. Consider the simple case in which there is no function in $\mathcal{H}_n^{\mathcal{O}}$ that is in the ε-neighborhood of the true function. Then, clearly, the probability that $\mathcal{O}$ outputs a hypothesis in the ε-neighborhood of the true function is 0.

Thus, in order for there to exist an Occam algorithm for a hypothesis class, there needs to be some kind of guarantee that, as n grows, there will always be some $g \in \mathcal{H}_n^{\mathcal{O}}$ that is within the ε-neighborhood of the true function. In fact, we might want the stronger condition that the true $f \in \mathcal{H}_n^{\mathcal{O}}$, if we want the probabilistic bounds to satisfy the PAC definition. This is a strong condition, and tacitly assuming it is what is doing all the work in the proof of theorem 1.

Now, it seems likely that a proponent of this justification of Occam's Razor might concede that it is true this assumption is needed. However, she might say, this is fine because this assumption is merely what it means for there to exist an Occam algorithm for a hypothesis space. In fact, I am aiding in the overall project of justifying Occam's Razor through PAC learning by more clearly stating the conditions under which an Occam algorithm exists. On first pass this seems reasonable. However, notice the nature of this condition. First, it makes specific reference to the true hypothesis f, which we are not supposed to know beforehand. Thus, this condition is not a condition of the hypothesis class but rather a condition about how $\mathcal{H}_n^{\mathcal{O}}$ relates to the true hypothesis at each n. However, as we saw in section 4.1, PAC learnability is defined for the family of hypothesis spaces and does not make reference to the particular true hypothesis. In fact, PAC learnability is a good definition because it does not depend on which hypothesis ends up being the true one—we can learn any hypothesis in any space belonging to the family.

In order to save the use of theorem 1, the proponent must reply with something along these lines: You are right that the condition must be expressed in terms of the family of spaces and should not depend on which hypothesis ends up being the true one. However, this is fine—we can see that an Occam algorithm exists for a class if, whichever hypothesis ends up being the true one, $\mathcal{H}_n^{\mathcal{O}}$ always includes a hypothesis in the ε-neighborhood of the true f. Clearly if $f \in \mathcal{H}_n^{\mathcal{O}}$ then this is satisfied. However, even if $\mathcal{H}_n^{\mathcal{O}}$ does not include f, all we require is that it includes some $g \in \mathcal{H}_n^{\mathcal{O}}$ that is within the ε-neighborhood of f.

https://doi.org/10.1086/709786 Published online by Cambridge University Press

I have two replies to the proponent. The first is weaker but I think worth stating; however, I take the second to be fatal for a justification of Occam's Razor of this sort. Before stating my replies, I want to clarify what this condition is. This seems like some kind of strong similarity condition of the hypothesis space (for each n). The functions of $\mathcal{H}_n$ must be sufficiently similar that, for every $h \in \mathcal{H}_n$, $\mathcal{H}_n^{\mathcal{O}}$ contains some g in the ε-neighborhood of h.

First, this is an incredibly strong condition, and if we take something like this to be an interesting condition for epistemic principles, then we can prove a great deal. Consider, for example, a principle we will call Super-Occam's Razor. Super-Occam's Razor tells us to consider only the simplest hypothesis (in terms of description length). Thus, $\mathcal{H}_n^{S\mathcal{O}}$ is set with only one element: the simplest hypothesis in the class. I can prove amazing bounds with this if we make the same kind of similarity assumption. If we assume that $\mathcal{H}_n^{S\mathcal{O}}$ contains a hypothesis in the ε-neighborhood of the true f, then, in fact, I do not need to see any examples; my algorithm just outputs the one hypothesis under consideration. Thus, I can construct a proof similar to that in appendix A and prove Super-Occam's Razor. Clearly, this cannot be what we have in mind when we are wondering whether a hypothesis class is learnable; we have imported too much information about the true hypothesis and its relation to others in the class to make it interesting from an epistemic point of view. Furthermore, such a result certainly does not provide a justification of the epistemic principle Super-Occam's Razor.

Second, let us grant the proponent that this is still interesting and, thus, that theorem 1 might still be able to provide an epistemic justification of Occam's Razor. One of the main reasons an Occam algorithm can learn efficiently as stated in theorem 1 is because of the similarity condition described above. However, this clearly has nothing to do with simplicity; this is why we can prove theorem 2, which makes use of complexity, with the same kind of (unstated) similarity condition. Both theorems rely on some way of choosing subsets of the hypothesis space that grow at the right speed with n—this is baked right into the definition of both an Occam algorithm and an Anti-Occam algorithm. As long as the similarity condition holds, these kind of "choose a particular subset that grows at the right rate" algorithms can learn.

There is nothing about simplicity in theorem 1 that is doing the work. What is really doing the work is a similarity condition; tracking the simplicity (or complexity) of hypotheses is inessential. With such a strong condition, different ways of taking subsets of hypothesis spaces that grow at a certain rate can achieve the same learning bounds, without relying on any notions of simplicity or complexity. Thus, theorem 1 cannot be a justification of Occam's Razor.

5. Other Concerns. Before concluding with a few remarks, I briefly put forward two further concerns about the result.

https://doi.org/10.1086/709786 Published online by Cambridge University Press

5.1. Encoding. In section 4, I did not challenge any of the assumptions or choices made in the proof of Occam's Razor. Instead, I argued against the interpretation of theorem 1 as a justification of Occam's Razor and took for granted the explicit background assumptions that led to these theorems. This revealed implicit assumptions of the argument.

In this section, however, I want to cast doubt on one of the main assumptions at play in the proof: the assumption that the length of a hypothesis's "standard encoding" is a good measure of the complexity of that hypothesis. I am not the first to notice this. In an earlier critique of the PAC-learning argument for Occam's Razor, Domingos writes: "[The result states] if a model with zero training-set error is found within a sufficiently small set of models, it is likely to have low generalization error. This model, however, could be arbitrarily complex. The only connection of this result to Occam's razor is provided by the information-theoretic notion that, if a set of models is small, its members can be distinguished by short codes" (1999, 410). Domingos is noting the tenuous link between the cardinality of $\mathcal{H}_n^{\mathcal{O}}$ and the coding. He also notes that the coding need not track the complexity of the model (what we have been calling "hypothesis").

I make two further objections to using encoding as a measure of complexity. The first, and more cautious, is that the notion of complexity is dependent on some standard fixed encoding. As Blumer et al. write, "the complexity of a hypothesis $h \in H$ [is] defined as the number of bits needed to represent h in some fixed encoding of the hypothesis in H" (1987, 379). I mentioned this earlier in section 3.1, in which I more formally defined complexity in this spirit using a mapping function ξ. Although the specific encoding is lost in the background of the presentation of proofs of Occam's Razor in the literature,[7] it is important to remember that, under this formulation, the complexity of a hypothesis is completely relative to the choice of ξ.[8] Choose a different encoding, and the complexity of a hypothesis could (in most cases would) change. In general, for a hypothesis h and two different encodings ξ_1 and ξ_2, it is not always the case that $\lambda(\xi_1(h)) = \lambda(\xi_2(h))$. Thus, an Occam algorithm that uses some ξ_1 as its encoding and one that uses some ξ_2 as its encoding might select entirely different hypotheses for consideration. In fact, ξ_2 might assign the shortest names to the most complex hypotheses under the ξ_1 encoding. Thus, even here it is hard to see how simplicity in the spirit of Occam is doing anything, since it is entirely relative to the choice of encoding. One might argue that there will be some more or less natural

7. Often it is mentioned in just one line with significantly less care than in sec. 3.1.

8. Note that moving from this simpler way of defining complexity to something more sophisticated like Kolmogorov complexity does not address the issue; the Kolmogorov complexity of a given hypothesis will still depend on how we choose to encode that hypothesis as a string.

https://doi.org/10.1086/709786 Published online by Cambridge University Press

encoding that does track a more objective notion of complexity. However, in this case no such argument or encoding was given—it was merely assumed (Blumer et al. 1987). As theorem 1 holds for any choice of encoding, it seems that we would need to have a quite different justification of Occam's Razor even if there were a standard encoding—the specific encoding would be doing none of the work.

This leads me to my second objection—that description length is a good measure of complexity at all. First, my concern above shows that description length (or "minimum description length") is not an absolute notion but is relative to the choice of encoding. Presumably, the intuitive notion of the complexity of a hypothesis is not dependent on the choice of language in which it is expressed. Even if there were some "standard" encoding, there is no reason to suppose that the length of the name of a hypothesis captures its complexity in the informal, intuitive sense. When one introduces a formalization of complexity—say, description length—and then moves from an epistemic principle that references this formal notion (e.g., a proof of the predictive virtue of shorter description length in a particular context) to one that references an informal notion like the general principle of Occam's Razor ("under very general assumptions, Occam's Razor produces hypotheses that with high probability will be predictive of future observations"; Blumer et al. 1987, 378), one is overstating the reach of one's result. It is against this that I want to caution: equivocating our informal notion of "simplicity" of a theory with a more formal notion such as description length.[9]

5.2. Direction of Implication. This section is an instance of the philosopher's old concern about necessary and sufficient conditions. Thus, I take my concern here not to be a devastating or original observation but rather a reminder that, even if the standard interpretation of theorem 1 had been largely correct, we would still be further away from a justification of Occam's Razor than it might seem before reflection.

As stated, theorem 1 says (informally) that any Occam algorithm for a family of hypothesis spaces is a good (PAC) learner. This is supposed to be a justification of Occam's Razor. However, even if there were no hidden assumptions that take away the force of the result (as I argued in sec. 4.2), theorem 1 is not enough to justify Occam's Razor as an epistemic principle. Even if it were straightforwardly true that any Occam algorithm for a class can learn that class efficiently, this does not give us reason to use Occam's Razor in general, for it is not guaranteed to work whenever (or even most of the time) a class is learnable. This much is made clear by theorem 2; if

9. Romeijn (2017) gives an excellent philosophical analysis of how formal complexity can come apart from what he calls "inherent" complexity in certain contexts.

https://doi.org/10.1086/709786 Published online by Cambridge University Press

theorem 1 did justify Occam's Razor, then theorem 2 would justify Anti-Occam's Razor. We cannot consistently have a justification for both razors from these formal results.

In order to justify Occam's Razor as an epistemic principle within this kind of framework, we would want a result that says that whenever a class is PAC learnable then an Occam algorithm can learn it. This is in fact slightly weaker than the other direction of implication of section 3.1, which would say that every PAC learner for a class is an Occam algorithm. Although there has been some attempt to prove a result like this, neither the direct converse of theorem 1 nor the weaker version has been proved.[10]

6. Conclusion. In this article I considered an argument from the PAC learning literature in favor of Occam's Razor as an epistemic principle. Bringing forward a number of objections, I argued that the standard interpretation of the result in the literature is misguided. I further argued that a better reading does not, in fact, support Occam's Razor at all.

Although this is only one argument for Occam's Razor, I am skeptical that we will be able to provide a solid epistemic justification of Occam's Razor.[11] Thus, my own suspicion is that Occam's Razor is merely a pragmatic way to make it easier for us to work with our theories.

Although PAC learning did not provide a justification for this particular epistemic principle, I think that philosophers, and formal epistemologists in particular, should care about the feasibility of the recommendations we seek to give to rational agents. Abstracting away from the resources and capabilities required by a method can be a useful idealization, but it does not paint a full picture of how agents should learn. In order to make ourselves more honest about the rational recommendations we give to agents, we should consider the limitations of the agents to which we give them. PAC learning is one such attempt to sharpen notions of learnability by incorporating computational complexity into the considerations of epistemic principles. The integration of the tools and concerns of philosophers and computer scientists will prove fertile for advancing our understanding of rational learning.

10. One of the closest results of which I am aware is by Moran and Yehudayoff (2015), in which they prove that if a set $C \subseteq \{0,1\}^X$ has Vapnik-Chervonenkis dimension d, then there is a sample compression scheme of size $2^{O(d)}$. Although this is certainly an interesting result, it does not get us the direction of implication that we would need for a justification of Occam's Razor along these lines.

11. It seems that whenever we try to sharpen notions of complexity or simplicity we find that either the justification fails (e.g., Solomonoff induction) or the notion of complexity we end up with looks very little like anything we would intuitively call complexity (notions that are dependent on the choice of encoding). An open candidate is Kelly's (2004) argument from truth-finding efficiency, but an evaluation of this framework is beyond the scope of this article.

https://doi.org/10.1086/709786 Published online by Cambridge University Press

Appendix A

Proof of Occam's Razor

The proof will follow Blumer et al. (1987) closely.

Lemma A1. Given a hypothesis space $\mathcal{H}_n$ and any $f \in \mathcal{H}_n$, the probability that any hypothesis in $\mathcal{H}_n$ with error larger than ε is consistent with m observations of f is less than $(1 - \varepsilon)^m|\mathcal{H}_n|$.

Proof. Let E_i be the event that hypothesis $h_i \in \mathcal{H}_n$ agrees with a single observation of f, and let E_i^m be the event that h_i agrees with all m observations of f. Let P^m be the product probability distribution of $\mathcal{X}^m$ according to the probability P. Let $\mathcal{D} = \{i : P(h_i \Delta f) > \varepsilon\}$. Then

$$P^m(\bigcup_{i \in \mathcal{D}} E_i^m)$$

is the probability of the event that there is some hypothesis in $\mathcal{H}_n$ that is outside of the ε-neighborhood of f (which I denote by $P(h_i \Delta f) > \varepsilon$) that agrees with m observations of f drawn according to the distribution P. It is this probability that we want to bound.

For any particular hypothesis $h_i \in \mathcal{H}_n$ outside the ε-neighborhood of f, the probability that it is consistent with a sample of size m drawn according to P is

$$P^m(E_i^m) < (1 - \varepsilon)^m$$

since each example is independently drawn according to P and the probability that h_i agrees with f on an example drawn according to P is less than $(1 - \varepsilon)$.

By the subadditivity of probability measures we have that

$$P^m(\bigcup_{i \in \mathcal{D}} E_i^m) \leq \sum_{i \in \mathcal{D}} P^m E_i^m$$

$$< \sum_{i \in \mathcal{D}} P^m E_i^m (1 - \varepsilon)^m \leq |\mathcal{H}_n|(1 - \varepsilon)^m.$$

Thus,

$$P^m(\bigcup_{i \in \mathcal{D}} E_i^m) < |\mathcal{H}_n|(1 - \varepsilon)^m,$$

which is our result. QED

The following proof of theorem 1 will follow Blumer et al. (1987) closely.

Proof. By lemma A1, there is an upper bound on the number of examples m needed. In particular, since we want the probability of outputting a

https://doi.org/10.1086/709786 Published online by Cambridge University Press

hypothesis outside the ε-neighborhood of the true f to be less than δ, we have that, if m satisfies

$$|\mathcal{H}_n|(1 - \varepsilon)^m \leq \delta,$$

then m is a sufficient number of examples.

Suppose there is some Occam algorithm $\mathcal{O}$ for a family $\mathcal{H} = \{\mathcal{H}_n : n \in N\}$ of hypothesis spaces. Since $\mathcal{O}$ is a learning algorithm, any hypothesis it outputs on input of a sample of size m will be consistent with all m observations.

To show that $\mathcal{O}$ produces with probability at least $1 - \delta$ a hypothesis within the ε-neighborhood of the true f with sample size m polynomial in ε, δ, n, we show that the following bound is sufficient:

$$m \geq \max\left\{\frac{2\ln(\delta^{-1})}{-\ln(1 - \varepsilon)}, \left(\frac{2p(n)\ln(2)}{-\ln(1 - \varepsilon)}\right)^{1/(1-\alpha)}\right\}.$$

We can rewrite the second lower bound as

$$p(n)m^{\alpha}\ln(2) \leq -\frac{m}{2}\ln(1 - \varepsilon).$$

Here we use the fact that $\mathcal{O}$ is an Occam algorithm. The algorithm only considers a hypothesis h if its complexity $\ell(h)$ is less than or equal to $p(n)m^{\alpha}$. Since ξ is a binary encoding, this means that the algorithm considers at most $2^{p(n)m^{\alpha}} = |\mathcal{H}_n^{\mathcal{O}}|$ possible hypotheses. By the previous bound, we have that

$$2^{p(n)m^{\alpha}} \leq (1 - \varepsilon)^{-m/2}.$$

If we multiple both sides by $(1 - \varepsilon)^m$, we get

$$2^{p(n)m^{\alpha}}(1 - \varepsilon)^m \leq (1 - \varepsilon)^{m/2}.$$

But then by lemma A1 and the first lower bound on m,

$$P^m\left(\bigcup_{i \in \mathcal{D}} E_i^m\right) < \left|\mathcal{H}_n^{\mathcal{O}}\right|(1 - \varepsilon)^m \leq (1 - \varepsilon)^{m/2} \leq \delta.$$

QED

Appendix B

Proof of Anti-Occam's Razor

The proof here runs parallel to that given in appendix A.

https://doi.org/10.1086/709786 Published online by Cambridge University Press

Proof. By lemma A1, there is an upper bound on the number of examples m needed:

$$|\mathcal{H}_n|(1 - \varepsilon)^m \leq \delta.$$

Suppose there is some Anti-Occam algorithm $\mathcal{A}$ for a family $\mathcal{H} = \{\mathcal{H}_n : n \in N\}$ of hypothesis spaces. Since $\mathcal{A}$ is a learning algorithm, any hypothesis it outputs on input of a sample of size m will be consistent with all m observations.

To show that $\mathcal{A}$ produces with probability at least $1 - \delta$ a hypothesis within the ε-neighborhood of the true f with sample size m polynomial in ε, δ, n, we show that the following bound is sufficient:

$$m \geq \max\left\{\frac{2\ln(\delta^{-1})}{-\ln(1 - \varepsilon)}, \left(\frac{2p(n)\ln(2)}{-\ln(1 - \varepsilon)}\right)^{1/1-\alpha}\right\}.$$

We can rewrite the second lower bound as

$$p(n)m^{\alpha}\ln(2) \leq -\frac{m}{2}\ln(1 - \varepsilon).$$

Here we use the fact that $\mathcal{A}$ is an Anti-Occam algorithm. The algorithm only considers a hypothesis h if it is one of the top $2^{p(n)m^{\alpha}} = |\mathcal{H}_n^{\mathcal{A}}|$ complex hypotheses. By the previous bound, we have that

$$2^{p(n)m^{\alpha}} \leq (1 - \varepsilon)^{-m/2},$$

and thus

$$2^{p(n)m^{\alpha}}(1 - \varepsilon)^m \leq (1 - \varepsilon)^{m/2}.$$

But then by lemma A1 and the first lower bound on m,

$$P^m(\bigcup_{i \in \mathcal{D}} E_i^m) < \left|\mathcal{H}_n^{\mathcal{A}}\right|(1 - \varepsilon)^m \leq (1 - \varepsilon)^{m/2} \leq \delta.$$

QED

REFERENCES

Aaronson, Scott. 2013. "Why Philosophers Should Care about Computational Complexity." In *Computability: Turing, Gödel, Church, and Beyond*, ed. B. Jack Copeland, Carl J. Posy, and Oron Shagrirj, 261–328. Cambridge, MA: MIT Press.

Blumer, Anselm, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. 1987. "Occam's Razor." *Information Processing Letters* 24 (6): 377–80.

Domingos, Pedro. 1999. "The Role of Occam's Razor in Knowledge Discovery." *Data Mining and Knowledge Discovery* 3 (4): 409–25.

Kearns, Michael J., and Umesh Virkumar Vazirani. 1994. *An Introduction to Computational Learning Theory*. Cambridge, MA: MIT Press.

https://doi.org/10.1086/709786 Published online by Cambridge University Press

Kelly, Kevin T. 2004. "Justification as Truth-Funding Efficiency: How Ockham's Razor Works." *Minds and Machines* 14 (4): 485–505.
Ming, Li, John Tromp, and Paul Vitányi. 2003. "Sharpening Occam's Razor." *Information Processing Letters* 85 (5): 267–74.
Ming, Li, and Paul Vitányi. 1990. "Kolmogorov Complexity and Its Applications." In *Algorithms and Complexity*, ed. Jan van Leeuwen, 187–254. Amsterdam: Elsevier.
Moran, Shay, and Amir Yehudayoff. 2015. "Proper PAC Learning Is Compressing." Electronic Colloquium on Computational Complexity Report no. 40, University of Trier.
Romeijn, Jan-Willem. 2017. "Inherent Complexity: A Problem for Statistical Model Evaluation." *Philosophy of Science* 84 (5): 797–809.
Sober, Elliott. 2015. *Ockham's Razors*. Cambridge: Cambridge University Press.
Sterkenburg, Tom F. 2016. "Solomonoff Prediction and Occam's Razor." *Philosophy of Science* 83 (4): 459–79.
Valiant, Leslie G. 1984. "A Theory of the Learnable." *Communications of the ACM* 27 (11): 1134–42.
———. 2013. *Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World*. New York: Basic.
Wolf, R. M., de. 1997. "Philosophical Applications of Computational Learning Theory: Chomskyan Innateness and Occzam's Razor." Master's thesis, Erasmus University.

https://doi.org/10.1086/709786 Published online by Cambridge University Press