Journal of Mathematical Psychology 76 (2017) 13–24

Contents lists available at ScienceDirect

Journal of Mathematical Psychology

journal homepage: www.elsevier.com/locate/jmp

Identification of probabilities

Paul M.B. Vitányi [a,b], Nick Chater [c,*]

[a] National Research Institute for Mathematics and Computer Science, CWI, Science Park 123, 1098 XG, Amsterdam, The Netherlands
[b] University of Amsterdam, The Netherlands
[c] Behavioural Science Group, Warwick Business School, University of Warwick, Coventry, CV4 7AL, UK

HIGHLIGHTS

- A fundamental problem of Bayesian inference is solvable in a number of contexts.
- Computability assumptions turn out crucially to simplify the learning problem.
- Exceptions can be learned from positive data, a long-standing puzzle in language acquisition.
- Data alone is often sufficient to learn an underlying model in perception.

ARTICLE INFO

Article history:
Received 1 August 2015
Received in revised form
13 October 2016
Available online 26 December 2016

Keywords:
Learning
Bayesian brain, identification
Computable probability
Markov chain
Computable measure
Typicality
Strong law of large numbers
Martin-Löf randomness
Kolmogorov complexity

ABSTRACT

Within psychology, neuroscience and artificial intelligence, there has been increasing interest in the proposal that the brain builds probabilistic models of sensory and linguistic input: that is, to infer a probabilistic model from a sample. The practical problems of such inference are substantial: the brain has limited data and restricted computational resources. But there is a more fundamental question: is the problem of inferring a probabilistic model from a sample possible even in principle? We explore this question and find some surprisingly positive and general results. First, for a broad class of probability distributions characterized by computability restrictions, we specify a learning algorithm that will almost surely identify a probability distribution in the limit given a finite i.i.d. sample of sufficient but unknown length. This is similarly shown to hold for sequences generated by a broad class of Markov chains, subject to computability assumptions. The technical tool is the strong law of large numbers. Second, for a large class of dependent sequences, we specify an algorithm which identifies in the limit a computable measure for which the sequence is typical, in the sense of Martin-Löf (there may be more than one such measure). The technical tool is the theory of Kolmogorov complexity. We analyze the associated predictions in both cases. We also briefly consider special cases, including language learning, and wider theoretical implications for psychology.

1. Introduction

Bayesian models in psychology and neuroscience postulate that the brain learns a generative probabilistic model of a set of perceptual or linguistic data (Chater, Tenenbaum, & Yuille, 2006; Oaksford & Chater, 2007; Pouget, Beck, Ma, & Latham, 2013; Tenenbaum, Kemp, Griffiths, & Goodman, 2011). Learning is therefore often viewed as an inverse problem. Some aspect of the world is presumed to contain a probabilistic model, from which data is sampled; the brain receives a sample of such data, e.g., at its sensory surfaces, and has the task of inferring the probabilistic model. That is, the brain has to infer an underlying probability distribution, from a sample from that distribution.

This theoretical viewpoint is implicit in a wide range of Bayesian models in cognitive science, which capture experimental data across many domains, from perception, to categorization, language, motor control, and reasoning (e.g., Chater & Oaksford, 2008). It is, moreover, embodied in a wide range of computational models of unsupervised learning in machine learning, computational linguistics, computer vision (e.g., Ackley, Hinton, & Sejnowski, 1985; Manning & Klein, 2003; Yuille & Kersten, 2006). Finally, the view that the brain recovers probabilistic models from sensory data is both theoretically prevalent and has received considerable empirical support in neuroscience (Knill & Pouget, 2004).

* Corresponding author.
E-mail addresses: paulv@cwi.nl (P.M.B. Vitányi), Nick.Chater@wbs.ac.uk (N. Chater).

http://dx.doi.org/10.1016/j.jmp.2016.11.004

The idea that the brain may be able to recover a probabilistic process from a sample of data from that process is an attractive one. For example, a recovered probabilistic model might potentially be used to explain past input or to predict new input. Moreover, sampling new data from the recovered probabilistic model could be used in the generation of new data from that probabilistic process, for creating mental images (Shepard, 1984) or producing language (Chater & Vitányi, 2007). Thus, from a Bayesian standpoint, one should expect that the ability to perceive should go alongside the ability to create mental images; and the ability to understand language should go alongside the ability to produce language. Thus, the Bayesian approach is part of the broader psychological tradition of analysis-by-synthesis, for which there is considerable behavioral and neuroscientific evidence in perceptual and linguistic domains (Pickering & Garrod, 2013; Yuille & Kersten, 2006).

Yet, despite its many attractions, the proposal that the brain recovers probabilistic processes from samples of data faces both practical and theoretical challenges. The practical challenges include the fact that the available data may be limited (e.g., children learn the probabilistic model of highly complex language using only millions of words). Moreover, the brain faces severe computational constraints: even the limited amount of data encountered will be encoded imperfectly and may rapidly be lost (Christiansen & Chater, 2016; Haber, 1983). The brain has limited processing resources to search and test the vast space of possible probabilistic models that might generate the data available.

In this paper we explore the conditions under which exactly inferring a probabilistic process from a stream of data is possible even in principle, with no restrictions on computational resources like time or storage or availability of data. If it turns out that there is no algorithm that can learn a probabilistic structure from sensory or linguistic experience when no computational or data restrictions are imposed, then this negative result will still hold when more realistic settings are examined.

Our analysis differs from previous approaches to these issues by assuming that the probabilistic process to be inferred is, in a way that will be made precise later, computable. Roughly speaking, the assumption is that the data to be analyzed is generated by a process that can be modeled by a computer (e.g., a Turing machine or a conventional digital computer) combined with a source of randomness (for example, a fair coin that can generate a limitless stream of random 0s and 1s that could be fed into the computer). There are three reasons to suppose that this focus on computable processes is interesting and not overly restrictive. First, some influential theorists have argued that all physical processes are computable in this, or stricter, senses (e.g., Deutsch, 1985). Second, most cognitive scientists assume that the brain is restricted to computable processes, and hence can only *represent* computable processes (e.g., Rescorla, 2015). According to this assumption, if it turns out that some aspects of the physical world are uncomputable, these will trivially be unlearnable simply because they cannot be represented; and, conversely, all aspects of learning of relevance to psychology, i.e., all aspects of the world that the brain can successfully learn, will be within the scope of our analysis. Third, all existing models of learning in psychology, statistics and machine learning are computable (and, indeed, are actually implemented on digital computers) and fall within the scope of the present results.

1.1. Background: pessimism about learnability

Within philosophy of science, cognitive science, and formal learning theory, a variety of considerations appear to suggest that negative results are likely. For example, in the philosophy of science it is often observed that theory is underdetermined by data (Duhem, 1914–1954; Quine, 1951): that is, an infinite number of theories is compatible with any finite amount of data, however large. After all, these theories can all agree on any finite data set, but diverge concerning any of the infinitely large set of possible data that has yet to be encountered. This might appear to rule out identifying the correct theory—and hence, *a fortiori* identify a correct probability distribution.

Cognitive science inherits such considerations, to the extent that the learning problems faced by the brain are analogous to those of inferring scientific theories (e.g., Gopnik, Meltzoff, & Kuhl, 1999). But cognitive scientists have also amplified these concerns, particularly in the context of language acquisition. Consider, for example, the problem of acquiring language from positive evidence alone, i.e., from hearing sentences of the language, but with no feedback concerning whether the learner's own utterances are grammatical or not (so-called negative evidence). It is often assumed that this is, to a good approximation, the situation faced by the child. This is because some and perhaps all children receive little useful feedback on their own utterances and ignore such feedback even when it is given (Bowerman, 1988). Yet, even without negative evidence, children nonetheless learn their native language successfully. For example, an important textbook on language acquisition (Crain & Lillo-Martin, 1999) repeatedly emphasizes that the child cannot learn *restrictions* on grammatical rules from experience—and that these must therefore somehow arise from innate constraints. For example, the English sentences *which team do you want to beat*, *which team do you wanna beat*, and *which team do you want to win*, would seem naturally to imply that **which team do you wanna win* is also a grammatical sentence. As indicated by the asterisk, however, this sentence is typically rejected as ungrammatical by native speakers. According to classical linguistic theory (e.g., Chomsky, 1982), the contraction to *wanna* is not possible because it is blocked by a "gap" indicating a missing subject—a constraint that has sometimes been presumed to follow from an innate universal grammar (Chomsky, 1980).

The problem with learning purely from positive evidence is that an overgeneral hypothesis, which does not include such restrictions, will be consistent with new data; given that languages are shot through with exceptions and restrictions of all kinds, this appears to provide a powerful motivation for linguistic nativism (Chomsky, 1980). But this line of argument cannot be quite right, because many exceptions are entirely capricious and could not possibly follow from innate linguistic principles. For example, the grammatical acceptability of *I like singing*, *I like to sing*, and *I enjoy singing* would seem to imply, wrongly, the acceptability of **I enjoy to sing*. But the difference between the distributional behavior of the verbs *like* and *enjoy* cannot stem from any innate grammatical principles. The fact that children are able to learn restrictions of this type, and the fact that they are so ubiquitous throughout language, has even led some scholars to speak of the *logical* problem of language acquisition (Baker & McCarthy, 1981; Hornstein & Lightfoot, 1981).

Similarly, in learning the meaning of words, it is not clear how, without negative evidence, the child can successfully retreat for overgeneralization. If the child initially proposes that, for example, *dog* refers to any animal, or that *mummy* refers to any adult female, then further examples will not falsify this conjecture. In word learning and categorization, and in language acquisition, researchers have suggested that one potential justification for overturning an overgeneral hypothesis is that absence-of-evidence can sometimes be evidence-of-absence (Hahn & Oaksford, 2008; Hsu, Horng, Griffiths, & Chater, 2016). That is, a child might take the absence of people using the word *dog* when referring to cats or mice; and the absence of *Mummy* being used to refer to other female friends or family members might lead to the child to be in doubt concerning their liberal use of these terms. But, of course, this line

of reasoning is not straightforward—for example, when learning *any* category that may apply in an infinite number of situations, the overwhelming majority of these will not have been encountered. It is not immediately clear how the child can tell the difference between benign, and genuinely suspicious, absence of evidence. The present results show that there is an algorithm that, under fairly broad conditions, can deal successfully with overgeneralization with probability 1, given sufficient data and computation time.

Previous results in the formal analysis of language learnability have reached more pessimistic conclusions, using different assumptions (Gold, 1967; Jain, Osherson, Royer, & Sharma, 1999). For example, as quoted in Pinker (1979), the pioneer of formal learning theory E. M. Gold points that "the problem with [learning only from] text is that if you guess too large a language, the sample will never tell you you're wrong" (Gold, 1967, p. 461). This is true if we allow very few assumptions about the structure of the text—and indeed negative results in this area frequently depend on demonstrating the existence of texts (i.e., samples of the language) with rather unnatural behavior precisely designed to mislead any putative learner. We shall see below that realistic, though still quite mild, assumptions, are sufficient to yield the opposite conclusion: that probability distributions, including probability distributions over languages, can be identified from positive instances alone.

1.2. Preview and examples

Consider, first, the case of independent, identical draws from a probability distribution. In many areas of psychology, the learning task is viewed as abstracting some pattern from a series of independent trials rather than picking up sequential regularities (although the i.i.d. assumption is not necessarily explicit). The i.i.d. case is relevant to problems as diverse as classical conditioning (Rescorla & Wagner, 1972, where a joint distribution between conditioned and unconditioned stimuli must be acquired) category learning (Shepard, Hovland, & Jenkins, 1961, where a joint distribution of category instances and labels is the target), artificial grammar learning or artificial language learning (Reber, 1989; Saffran, Aslin, & Newport, 1996, where a probability distribution over strings of letters or sounds is to be learned). Similarly, the i.i.d. assumption is often implicit in learning algorithms in cognitive science and machine learning, such as, for example, many Bayesian and neural network models in perception, learning and categorization (e.g., Ackley et al., 1985).

Learning such potentially complex patterns from examples may seem challenging. Yet even analyzing perhaps the simplest case, learning the probability distribution of a biased coin is not straightforward. For concreteness, consider flipping a coin, with probability p of coming up heads. Suppose that we can flip the coin endlessly, and can, at every point as the sequence of data emerges, guess the value of p; we can change our mind as often as we like. It is natural to wonder whether there is some procedure for guessing such that, after some point, we stick to our guess—and that this guess is, either certainly or with high probability, correct. So, for example, if the coin is a fair coin, such that $p = 0.5$, can we eventually lock on to the conjecture that the coin is fair and, after some point, never change this conjecture however much data we receive?

The answer is by no means obvious, even for such simple case. After all, the difference between the number of heads and tails will fluctuate, and can grow arbitrarily large—and such fluctuations might persuade us, wrongly, that the coin is biased in favor, or against, heads. How sure can we be that, eventually, we will successfully identify the precise bias of a coin that *is* biased, e.g., where $p = 3/4$ or $p = 1/3$?

Or, to step up the level of complexity very considerable, consider the problem of inferring a stochastic phrase structure grammar from an indefinitely large sample of i.i.d. sentences generated from that grammar.[1] Or suppose the input is a sequence of images generated drawn from a probabilistic image model such as a Markov random field—can a perceiver learn to precisely identify the probabilistic model of the image, given sufficient data?

As we shall see in Section 3, remarkably, it turns out that, with fairly mild restrictions (a restricted computability), with probability 1, it is possible to infer in the limit, the correct probability distribution exactly, given a sufficiently large finite supply of i.i.d. samples. Moreover, it is possible to specify a computable algorithm that will reliably find this probability distribution. A similar result holds for ergodic Markov chains, which broadens its application considerably.

This result is unexpectedly strong, given mild restrictions on computability (which we describe in detail below). In particular, it shows that there is no *logical* problem concerning the possibility of learning languages, or other patterns, which contain exceptions, from positive evidence alone. As noted above, it has been influentially argued in linguistics and the study of language acquisition that exceptions (examples that are *not* possible) cannot be learned purely by observing their non-occurrence, because there are, after all, infinitely many linguistic forms which are possible but also have not been observed (e.g., Crain & Lillo-Martin, 1999). A variety of arguments and results have suggested that, despite such arguments, languages with exceptions can be learned successfully (Chater, Clark, Goldsmith, & Perfors, 2015; Clark & Lappin, 2010; Pullum & Scholz, 2002).

The present result shows that with the mentioned restrictions, given i.i.d. data it is possible exactly to learn the probability distribution of languages from a sample; or, from Markovian outputs, it is possible exactly to learn the Markov chain involved. An earlier result in Chater and Vitányi (2007) showed that language acquisition with sufficient data was possible on the assumption that an ideal learner could find the shortest description of a corpus. But finding the shortest description is known to be uncomputable. By contrast, the present paper focuses on what can be learned by a computable learner, provides an explicit algorithm by which that learner can operate, and considers exact learning rather than approximating the language arbitrarily accurately.

We also consider what can be learned if we weaken the i.i.d. restriction (and the mentioned Markov chain restriction) considerably—to deal with the possibility of learning sequential data that is generated by a computable process (we make this precise below). Many aspects of the environment, from the flow of visual and auditory input, to the many layers of sequential structure relating successive sentences, paragraphs, and chapters, while reading a novel, are not well approximated by identical independent sampling from a fixed distribution or the output of a small Markov chain. Nonetheless, the brain appears to be able to discover their structure, at least to some extent, with remarkable effectiveness.

One particularly striking illustration of the power to predict subsequent input is Shannon's method for estimating the entropy of English (Shannon, 1951). Successively predicting the next letter in a text, given previous letters, one or two guesses often suffice, leading to the conclusion that English texts typically can be encoded using little more than one bit of information per letter (while more than four bits would be required if the 26 letters were treated

[1] A stochastic phrase structure grammar is a conventional phrase structure grammar, with probabilities associated with each of the rewrite rules. For example, a noun phrase might sometimes expand to give a determiner followed by a noun, while sometimes expanding to give a single proper noun; and individual grammatical categories, such as proper nouns, map probabilistically on specific proper nouns.

as occurring independently). The ability to predict incoming sequential input is, of course, important for reacting to the physical or linguistic environment successfully, by predicting dangers and opportunities and acting accordingly. Many theorists also see finding structure in sequential material as fundamental to cognition and learning (Clark, 2013; Elman, 1990; Hollerman & Schultz, 1998; Kilner, Friston, & Frith, 2007).

If we weaken the i.i.d. or above Markovian assumption, what alternative restriction on sequential structure can we impose, and still obtain tractable analytical results? Clearly if there are no restrictions on structure of the process at all, then there are no constraints between prior and subsequent material. It turns out, though, a surprisingly minimal restriction is sufficient: we assume, roughly, only that the sequential material is generated by a mildly restricted *computable* dependent probabilistic process (this will be made precise below). Unlike the i.i.d. or Markov case, different such processes could have generated this sample; but it turns out that, given a finite sample that is long enough and that is guaranteed to be the initial segment of an infinite typical output of one of those computable dependent probabilistic processes, it is possible to infer a single process exactly (out of a number of such processes) according to which that sample is an initial segment of an infinite typical sample. We shall discuss these issues in Section 4.

Throughout this paper, we focus on learning probabilities themselves, rather than particular representations of probabilities. If there is at least one computer program representing a function, there are, of course, infinitely many such programs (representing the data in slightly different ways, incorporating additional null operations, and so on). The same is true for programs representing probability distributions. For some purposes, these differences in representation may be crucial. For example, psychologists and linguists may be interested in which of an infinite number of equivalent grammars – from the point of view of the sentences allowed – is represented by the brain. But, from the point of view of the problem of learning, we must treat them as equivalent. Indeed, it is clear that no learning method from observations alone could ever distinguish between models which generate precisely the same probability distribution over possible observations.

Our discussion begins with an introduction of our formal framework, in the next section. We then turn to the case of i.i.d. draws from a computable mass function, and to runs of a computable ergodic Markov chain, using the strong law of large numbers as the main technical tool. The next section *Computable Measures* considers learning material with computable sequential dependencies; here the main technical tool is Kolmogorov complexity theory. We then briefly consider whether these results have implications for the problem of predicting future data, based on past data, before we draw brief conclusions. The mathematical details and detailed proofs are relegated to Appendices.

2. The formal framework

We follow in the general theoretical tradition of formal learning theory, where we abstract away from specific representational questions, and focus on the underlying abstract structure of the learning problem.

One can associate the natural numbers with a lexicographic length-increasing ordering of finite strings over a finite alphabet. A natural number corresponds to the string of which it is the position in the thus established order. Since a language is a set of sentences (finite strings over a finite alphabet), it can be viewed as a subset of the natural numbers. (In the same way, natural numbers could be associated with images or instances of a concept). The learnability of a language under various computational assumptions is the subject of an immensely influential approach in Gold (1965) and especially (Gold, 1967), or the review (Jain et al., 1999). But surely in the real world the chance of one sentence of a language being used is different from another. For example, in general short sentences have a larger chance of turning up than very long sentences. Thus, the elements of a given language are distributed in a certain way. There arises the problem of identifying or approximating this distribution.

Our model is formulated as follows: we are given a sufficiently long finite sequence of data consisting of elements drawn from the set (language) according to a certain probability, and the learner has to identify this probability. In general, however much data has been encountered, there is no point at which the learner can announce a particular probability as correct with certainty. Weakening the learning model, the learner might learn to identify the correct probability in the limit. That is, perhaps the learner might make a sequence of guesses, finally locking on to correct probability and sticking to it forever—even though the learner can never know for sure that it has identified the correct probability successfully. We shall consider identification in the limit (following, for example, Gold, 1967; Jain et al., 1999; Pinker, 1979). Since this is not enough we additionally restrict the type of probability.

In conventional statistics, probabilistic models are typically idealized as having continuous valued parameters; and hence there is an uncountable number of possible probabilities. In general it is impossible that a learner can make a sequence of guesses that precisely locks on to the correct values of continuous parameters. In the realm of algorithmic information theory, in particular in Solomonoff induction (Solomonoff, 1964) and here, we reason as follows. The possible strategies of learners are computable in the sense of Turing (1936), that is, they are computable functions. The set of these is discrete and thus countable. The hypotheses that can be learned are therefore countable and computable, and in particular the set of probabilities from which the learner chooses must be *computable*. Indeed, this argument can be interpreted as showing that the fundamental problem is one of representation: the overwhelming majority of real-valued parameters cannot be *represented* by any computable strategy; and hence *a fortiori* cannot possible be learned.

Our starting point is that it is only of interest to consider the identifiability of computable hypotheses—because hypotheses that are not computable cannot be represented, let alone learned. Making this precise requires specifying what it means for a probability distribution to be computable. Moreover, it turns out that computability is not enough, it is also necessary that the considered set of computable probabilities is computably enumerable (c.e.) or co-computable enumerable (co-c.e.), all of which are explained in Appendix A. Informally, a subset of a set is c.e. if there is a computer which enumerates all the elements of the subset but no element outside the subset (but in the set). For example, the computable probability mass functions (or computable measures) for which algorithms are known can be computably enumerated in lexicographic order of the algorithms. Hence they satisfy Theorem 1 (or Theorem 2). A subset is co-c.e. if all elements outside the subset (but in the set) can be enumerated by a computer. In our case the set comprises all computable probability mass functions, respectively, all computable measures. Since by Lemma 1 in Appendix A this set is not c.e., a subset that is c.e. (or co-c.e.) is a proper subset, that is, it does not contain all computable probability mass functions, respectively, all computable measures.

In the exposition below, we consider two cases. In case 1, the data are drawn independent identically distributed (i.i.d.) from a subset of the natural numbers according to a probability mass function in a c.e. or co-c.e. set of computable probability mass functions, or consist of a run of a member of a c.e. or co-c.e. set of computable ergodic Markov chains. For this case, there is, as we

have noted, a learning algorithm that will almost surely identify a probability distribution in the limit. This is the topic of Section 3.

In case 2 the elements of the infinite sequence are dependent and the data sequence is typical for a measure from a c.e. or co-c.e. set of computable measures. For this more general case, we prove a weaker, though still surprising result: that there is an algorithm which identifies in the limit a computable measure for which that sequence is typical (in the sense introduced by Martin-Löf). These results are the focus of Section 4.

2.1. Preliminaries

Let $\mathcal{N}$, $\mathcal{Q}$, $\mathcal{R}$, and $\mathcal{R}^+$ denote the natural numbers, the rational numbers, the real numbers, and the nonnegative real numbers, respectively. We say that we *identify* a function f *in the limit* if we have an algorithm which produces an infinite sequence $f_1, f_2, \ldots$ of functions and $f_i = f$ for all but finitely many i. This corresponds to the notion of "identification in the limit" in Gold (1967), Jain et al. (1999), Pinker (1979) and Zeugmann and Zilles (2008). In this notion at every step an object is produced and after a finite number of steps the target object is produced at every step. However, we do not know this finite number. It is as if you ask directions and the answer is "at the last intersection turn right", but you do not know which intersection is last. The functions f we want to identify in the limit are probability mass functions, Markov chains, or measures.

Definition 1. Let $L \subseteq \mathcal{N}$ and its associated *probability mass function* p a function $p : L \to \mathcal{R}^+$ satisfying $\sum_{x \in L} p(x) = 1$. A Markov chain is an extension as in Definition 2. A *measure* μ is a function $\mu : L^* \to \mathcal{R}^+$ satisfying the measure equalities in Appendix C.

2.2. Related work

In Angluin (1988) (citing previous more restricted work) a target probability mass function was identified in the limit when the data are drawn i.i.d. in the following setting. Let the target probability mass function p be an element of a list $q_1, q_2, \ldots$ subject to the following conditions: (i) every $q_i : \mathcal{N} \to \mathcal{R}^+$ is a probability mass function; (ii) we exhibit a computable total function $C(i, x, \epsilon) = r$ such that $q_i(x) - r \leq \epsilon$ with $r, \epsilon > 0$ are rational numbers. That is, there exists a rational number approximation for all probability mass functions in the list up to arbitrary precision, and we give a single algorithm which for each such function exhibits such an approximation. The technical means used are the law of the iterated logarithm and the Kolmogorov–Smirnov test. However, the list $q_1, q_2, \ldots$ cannot contain all computable probability mass functions because of a diagonal argument, Lemma 1.

In Barron and Cover (1991) computability questions are apparently ignored. The *Conclusion* states "If the true density [and hence a probability mass function] is finitely complex [it is computable] then it is exactly discovered for all sufficiently large sample sizes". The tool that is used is estimation according to $\min_q(L(q) + \log(1/\prod_{i=1}^n q(X_i)))$. Here q is a probability mass function, $L(q)$ is the length of its code and $q(X_i)$ is the q-probability of the ith random variable X_i. To be able to minimize over the set of computable q's, one has to know the $L(q)$'s. If the set of candidate distributions is countably infinite, then we can never know when the minimum is reached—hence at best we have then identification in the limit. If $L(q)$ is identified with the Kolmogorov complexity $K(q)$, as in Section 4 of this reference, then it is uncomputable as already observed by Kolmogorov in Kolmogorov (1965) (for the plain Kolmogorov complexity; the case of the prefix Kolmogorov complexity $K(q)$ is the same). Computable $L(q)$ (given q) cannot be computably enumerated; if they were this would constitute a computable enumeration of computable q's which is impossible by Lemma 1. To obtain the minimum we require a computable enumeration of the $L(q)$'s in the estimation formula. The results hold (contrary to what is claimed in the *Conclusion* of Barron and Cover (1991) and other parts of the text) not for the set of computable probability mass functions since they are not c.e. The sentence "you know but you don't know you know" on the second page of Barron and Cover (1991) does not hold for an arbitrary computable mass probability.

In reaction to an earlier version of this paper with too large claims as described in Appendix E, in Bienvenu, Monin, and Shen (2014) it is shown that it is impossible to identify an arbitrary computable probability mass function (or measure) in the limit given an infinite sequence of elements from its support (which sequence is guaranteed to be typical for some computable measure in the measure case).

2.3. Results

The set of halting algorithms for computable probabilities (or measures) is not c.e., Lemma 1 in Appendix A. This complicates the algorithms and analysis of the results. In Section 3 there is a computable probability mass function (the target) on a set of natural numbers. We are given a sufficiently long finite sequence of elements of this set that are drawn i.i.d. and are asked to identify the target. An algorithm is presented which identifies the target in the limit almost surely provided the target is an element of a c.e. or co-c.e. set of halting algorithms for computable probability mass functions (Theorem 1). This also underpins the result announced in Hsu, Chater, and Vitányi (2011, Theorem 1 in the Appendix and appeals to it in the main text of the reference) with the following modification "computable probabilities" need to be replaced by "c.e. and co-c.e. sets of computable probabilities". If the target is an element of a c.e. or co-c.e. set of computable ergodic Markov chains then there is an algorithm with as input a sequence of states of a run of the Markov chain and as output almost surely the target (Corollary 1). The technical tool is in both cases the strong law of large numbers. In Section 4 the set of natural numbers is also infinite and the elements of the sequence are allowed to be dependent. We are given a guarantee that the sequence is typical (Definition 4) for at least one measure from a c.e. or co-c.e. set of halting algorithms for computable measures. There is an algorithm which identifies in the limit a computable measure for which the data sequence is typical (Theorem 2). The technical tool is the Martin-Löf theory of sequential tests (Martin-Löf, 1966) based on Kolmogorov complexity. In Section 5 we consider the associated predictions, and in Section 6 we give conclusions. In Appendix A we review the used computability notions, in Appendix B we review notions of Kolmogorov complexity, in Appendix C we review the measure and computability notions that we use. We defer the proofs of the theorems to Appendix D. In Appendix E we give the tortuous genesis of the results.

3. Computable probability mass functions and i.i.d. drawing

To approximate a probability in the i.i.d. setting is well-known and an easy example to illustrate our problem. One does this by an algorithm computing the probability $p(a)$ in the limit for all $a \in L \subseteq \mathcal{N}$ almost surely given the infinite sequence $x_1, x_2, \ldots$ of data i.i.d. drawn from L according to p. Namely, for $n = 1, 2, \ldots$ for every $a \in L$ occurring in $x_1, x_2, \ldots, x_n$ set $p_n(a)$ equal to the frequency of occurrences of a in $x_1, x_2, \ldots, x_n$. Note that the different values of p_n sum to precisely 1 for every $n = 1, 2, \ldots$. The output is a sequence $p_1, p_2, \ldots$ of probability mass functions such that we have $\lim_{n\to\infty} p_n = p$ almost surely, by the strong law of large numbers (see Claim 1). The probability mass functions considered here consist of *all* probability mass functions

on L—computable or not. The probability mass function p is thus represented by an approximation algorithm.

In this paper we deal only with computable probability mass functions. If p is computable then it can be represented by a halting algorithm which computes it as defined in Appendix A. Most known probability mass functions are computable provided their parameters are computable. In order that it is computable we only require that the probability mass function is finitely describable and there is a computable process producing it (Turing, 1936).

One issue is how short the code for p is, a second issue is the computability properties of the code for p, and a third issue is how much of the data sequence is used in the learning process. The approximation of p above results in a sequence of codes of probabilities $p_1, p_2, \ldots$ which are lists of the sample frequencies in an initial finite segment of the data sequence. The code length of the list of frequencies representing p_n grows usually to infinity as the length n of the segment grows to infinity. The learning process involved uses all of the data sequence and the result is an encoding of the sample frequencies in the data sequence in the limit. The code for p is usually infinite. This holds as well if p is computable. Such an approximation contrasts with identification in the following.

Theorem 1 (*i.i.d. Computable Probability Identification*)**.** *Let L be a set of natural numbers and p be a probability mass function on L. This p is described by an element of a c.e. or co-c.e. set of halting algorithms for computable probability mass functions. There is an algorithm identifying p in the limit almost surely from an infinite sequence $x_1, x_2, \ldots$ of elements of L drawn i.i.d. according to p. The code for p via an appropriate Turing machine is finite. The learning process uses only a finite initial segment of the data sequence and takes finite time.*

We do not know how large the finite items in the theorem are. The proof of the theorem is deferred to Appendix D. The intuition is as follows. By assumption the target probability mass function is a member of a linear list of halting algorithms for computable probability mass functions listed as list $\mathcal{A}$. By the strong law of large numbers we can approximate the target probability mass function by the sample means. Since the members of $\mathcal{A}$ are linearly ordered we can after each new sample compute the least member which agrees best according to a certain criterion with the samples produced thus far. At some stage this least element does not change any more.

Example 1. Since the c.e. and co-c.e. sets strictly contain the computable sets, Theorem 1 is strictly stronger than the result in Angluin (1988) referred to in Section 2.2. It is also strictly stronger than Barron and Cover (1991) that does not give identification in the limit for classes of computable functions.

Define the primitive computable probability mass functions as the set of probability mass functions for which it is decidable that they are constructed from primitive computable functions. Since this set is computable it is c.e. The theorem shows that identification in the limit is possible for members of this set. Define the time-bounded probability mass functions for any fixed computable time bound as the set of elements for which it is decidable that they are computable probability mass functions satisfying this time bound. Since this set is computable it is c.e. Again, the theorem shows that identification in the limit is possible for elements from this set.

Another example is as follows. Let $L = \{a_1, a_2, \ldots, a_n\}$ be a finite set. The primitive recursive functions $f_1, f_2, \ldots$ are c.e. Hence the probability mass functions $p_1, p_2, \ldots$ on L defined by $p_i(a_j) = f_i(j)/\sum_{h=1}^{n} f_i(h)$ are also c.e. Let us call these probability mass functions simple. By Theorem 1 they can be identified in the limit. ◊

The class of probability mass functions for which the present result applies is very broad. Suppose, for example, that we frame the problem of language acquisition in the following terms: a corpus is created by i.i.d. sampling from some primitive recursive language generation mechanism (for example, a stochastic phrase structure grammar (Charniak, 1996) with rational probabilities, or an equivalent, but more cognitively motivated formalism such as tree-adjoining grammar (Joshi & Schabes, 1997) or combinatory categorical grammar (Steedman, 2000). That is, the algorithm described here will search possible programs which correspond to generators of grammars, and will eventually find, and never change from, a stochastic grammar that precisely captures the probability mass function that generated the linguistic data. That is, the present result implies that there is a learning algorithm that identifies in the limit the probability mass function according to which these sentences are generated with probability 1. Of course, there may, in general, within any reasonably rich stochastic grammar formalism, be many ways of representing the probability distribution over possible sentences (just as there are many computer programs that code for the same function). Of course, no learning process can distinguish between these, precisely because they are, by assumption, precisely equivalent in their predictions. Hence, an appropriate goal of learning can only be to find the underlying probability mass function, rather than attempting the impossible task of inferring the particular representation of that function.

The result applies, of course, not just to language but to learning structure in perceptual input, such as visual images. Suppose that a set of visual images is created by i.i.d. sampling from a Markov random field with rational parameters (Li, 2012); then there will be a learning algorithm which identifies in the limit the probability distribution over these images with probability 1. The result applies, also, to the unsupervised learning of environmental structure from data, for example by connectionist learning methods (Ackley et al., 1985) or by Bayesian learning methods (Chater et al., 2006; Pearl, 2014; Tenenbaum et al., 2011).

3.1. Markov chains

I.i.d. draws from a probability mass function is a special case of a run of a discrete Markov chain. We investigate which Markov chains have an equivalent of the strong law of large numbers. Theorem 1 then holds *mutatis mutandis* for these Markov chains. First we need a few definitions.

Definition 2. A sequence of random variables $(X_t)_{t=0}^{\infty}$ with outcomes in a finite or countable state space $S \subseteq \mathcal{N}$ is a *discrete time-homogeneous Markov chain* if for every ordered pair i, j of states the quantity $q_{i,j} = \Pr(X_{t+1} = j | X_t = i)$ called the *transition probability* from state i to state j, is independent of t. If M is such a Markov chain then its associated *transition matrix* Q is defined as $Q := (q_{i,j})_{i,j \in \mathcal{N}}$. The matrix Q is non-negative and its row sums are all unity. It is infinite dimensional when the number of states is infinite.

In the sequel we simply speak of "Markov chains" and assume they satisfy Definition 2.

Definition 3. A Markov chain M is *ergodic* if it has a stationary distribution $\pi = (\pi_x)_{x \in S}$ satisfying $\pi Q = \pi$ and for every distribution $\sigma \neq \pi$ holds $\sigma Q \neq \sigma$. This stationary distribution π satisfies $\pi_x > 0$ for all $x \in S$ and $\sum_{x \in S} \pi_x = 1$. With X_t being the state of the Markov chain at epoch t starting from $X_0 = x_0 \in S$ we have

$$\lim_{n \to \infty} \frac{1}{n} \sum_{t=1}^{n} X_t = \mathbf{E}_\pi[X] = \sum_{x \in S} \pi_x x, \tag{3.1}$$

approximating theoretical means by sample means. An ergodic Markov chain is *computable* if its transition probabilities and stationary distribution are computable.

Corollary 1 (*Identification Computable Ergodic Markov Chains*)**.** *Consider a c.e. or co-c.e. set of halting algorithms for computable ergodic Markov chains. Let M be an element of this set. There is an algorithm identifying M in the limit almost surely from an infinite sequence $x_1, x_2, \ldots$ of states of M produced by a run of M. The code for M via an appropriate Turing machine is finite. The learning process uses only a finite initial segment of the data sequence and takes finite time.*

Example 2. Let M be an ergodic Markov chain with a finite set S of states. There exists a unique distribution π over S with strictly positive probabilities such that

$$\lim_{s\to\infty} q_{i,j}^s = \pi_j,$$

for all states i and j. In this case we have that $\pi^0 Q^t \to \pi$ pointwise as $t \to \infty$ and the limit is independent of π^0. The stationary distribution π is the unique vector satisfying $\pi Q = \pi$, where $\sum_i \pi_i = 1$. (Necessary and sufficient conditions for ergodicity are that the chain should be *irreducible*, that is, for each pair of states i, j there is an $s \in \mathcal{N}$ such that $q_{i,j}^s > 0$ (state j can be reached from state i in a finite number of steps); and *aperiodic*, the $\gcd\{s : q_{i,j}^s > 0\} = 1$ for all $i, j \in T$.)

Equation $\pi Q = \pi$ is a system of N linear equations in N unknowns (the entries π_j). We can solve the unknowns by a computable procedure: in the first equation express one variable in terms of the others; substitute the expression into the remaining equations; repeat this process until the last equation; solve it and then back substitute until the total solution is found.

Since π is unique, the system of linear equations has a unique solution. If the original entries of Q are computable, then this process keeps the entries of π computable as well. Therefore, if the transition probabilities of the Markov chain are computable, then the stationary distribution π is a computable probability mass function. We now invoke the Ergodic Theorem approximating theoretical means by sample means (Feller, 1968; Lange, 2005) as in (3.1). ◊

4. Computable measures

In the i.i.d. case we dealt with a process where the future was independent of the present or the past, in the Markov case we extended this independence such that the immediate future is determined by the present but not by the past of too long ago. What can be shown if we drop the assumption of independence altogether? Then we go to measures as defined in Appendix C. As far as the authors are aware, for general measures there exists neither an approximation as in Section 3 nor an analog of the strong law of large numbers. However, there is a notion of typicality of an infinite data sequence for a computable measure in the Martin-Löf theory of sequential tests (Martin-Löf, 1966) based on Kolmogorov complexity, and this is what we use.

Let $L \subseteq \mathcal{N}$ and μ be a measure on L^∞ in a c.e. or co-c.e. set of halting algorithms for computable measures. In this paper, instead of the common notation $\mu(\Gamma_x)$ we use the simpler notation $\mu(x)$. We are given a sequence in L^∞ which is typical (Definition 4 in Appendix C) for μ. The constituent elements of the sequence are possibly dependent. The set of typical infinite sequences of a computable measure μ have μ-measure one, and each typical sequence passes all computable tests for μ-randomness in the sense of Martin-Löf. This probability model is much more general than i.i.d. drawing according to a probability mass function. It includes stationary processes, ergodic processes, Markov processes of any order, and many other models. In particular, this probability model includes many of the models used in mathematical psychology and cognitive science.

Theorem 2 (*Computable Measure Identification*)**.** *Let L be a set of natural numbers. We are given an infinite sequence of elements from L and this sequence is guaranteed to be typical for at least one measure in a c.e. or co-c.e. set of halting algorithms for computable measures. There is an algorithm which identifies in the limit (certainly) a computable measure in the c.e. or co-c.e set above for which the sequence is typical. The code for this measure is an appropriate Turing machine and is finite. The learning process takes finite time and uses only a finite initial segment of the data sequence.*

The proof is deferred to Appendix D.[2] We give an outline of the proof of Theorem 2. Let $\mathcal{B}$ be a list of a c.e. or co-c.e. set of halting algorithms for computable measures. Assume that each measure occurs infinitely many times in $\mathcal{B}$. For a measure μ in the list $\mathcal{B}$ define

$$\sigma(j) = \log 1/\mu(x_1 \ldots x_j) - K(x_1 \ldots x_j).$$

By (C.2) in Appendix C, data sequence $x_1, x_2, \ldots$ is typical for μ iff $\sup_j \sigma(j) = \sigma < \infty$. By assumption there exists a measure in $\mathcal{B}$ for which the data sequence is typical. Let μ_h be such a measure. Since halting algorithms for μ_h occur infinitely often in the list $\mathcal{B}$ there is a halting algorithm $\mu_{h'}$ in the list $\mathcal{B}$ with $\sigma_{h'} = \sigma_h$ and $\sigma_h < h'$. This means that there exists a measure μ_k in $\mathcal{B}$ for which the data sequence $x_1, x_2, \ldots$ is typical and $\sigma_k < k$ with k least.

Example 3. Let us look at some applications. Define the primitive recursive measures as the set of objects for which it is decidable that they are measures constructed from primitive recursive functions. Since this set is computable it is c.e. The theorem shows that identification in the limit is possible for primitive recursive measures.

Define the time-bounded measures for any fixed computable time bound as the set of objects for which it is decidable that they are measures satisfying this time bound. Since this set is computable it is c.e. Again, the theorem shows that identification in the limit is possible for elements from this set.

Let L be a finite set of cardinality l, and $f_1, f_2, \ldots$ be a c.e. set of the primitive recursive functions with domain L. Computably enumerate the strings $x \in L^*$ in lexicographical length-increasing order. Then every string can be viewed as the integer giving its position in this order. Let ϵ denote the *empty word*, that is, the string of length 0. Confusion with the notation ϵ equals a small quantity is avoided by the context. Define $\mu_i(\epsilon) = f_i(\epsilon)/f_i(\epsilon) = 1$, and inductively for $x \in L^*$ and $a \in L$ define $\mu_i(xa) = f_i(xa)/\sum_{a\in L} f_i(xa)$. Then $\mu_i(x) = \sum_{a\in L} \mu_i(xa)$ for all $x \in L^*$. Therefore μ_i is a measure. Call the c.e. set $\mu_1, \mu_2, \ldots$ the simple measures. The theorem shows that identification in the limit is possible for the set of simple measures. ◊

5. Prediction

In Section 3 the data are drawn i.i.d. according to an appropriate probability mass function p on the elements of L. Given p, we can predict the probability $p(a|x_1, \ldots, x_n)$ that the next draw results in an element a when the previous draws resulted in $x_1, \ldots, x_n$. (The resulting measure on L^∞ is called an i.i.d. measure.) Once we have identified p, prediction is possible (actually after a finite but unknown running time of the identifying algorithm). The same holds for ergodic Markov chains (Corollary 1). This is reassuring

[2] Theorems 2 and 1 are incomparable although it is tempting to think the latter is a corollary of the former. The infinite sequences considered in Theorem 2 are typical for some computable measure. Restricted to i.i.d. measures (the case of Theorem 1) such sequences are a proper subset from those resulting from i.i.d. draws from the corresponding probability mass function. This is the reason why the result of Theorem 2 is "certain" and the result from Theorem 1 is "almost surely".

for cognitive scientists and neuroscientists who see prediction as fundamental to cognition (Clark, 2013; Elman, 1990; Hollerman & Schultz, 1998; Kilner et al., 2007).

For general measures as in Section 4, allowing dependent data, the situation is quite different. We can meet the so-called black swan phenomenon of Popper (1959). Let us give a simple example. The data sequence is $a, a, \ldots$ is typical (Definition 4) for the measure μ_1 defined by $\mu_1(x) = 1$ for every data sequence x consisting of a finite or infinite string of a's and $\mu_1(x) = 0$ otherwise. But $a, a, \ldots$ is also typical for the measure μ_0 defined by $\mu_0(x) = \frac{1}{2}$ for every string x consisting of a finite or infinite string of a's, and $\mu_0(x) = \frac{1}{2}$ for a string x consisting of initially a fixed number n of a's followed by a finite or infinite string of b's, and 0 otherwise. Then, μ_1 and μ_0 give different predictions with an initial n-length sequence of a's. But given a data sequence consisting initially of only a's, a sensible algorithm will predict a as the most likely next symbol. However, if the initial data sequence consists of n symbols a, then for μ_1 the next symbol will be a with probability 1, and for μ_0 the next symbol is a with probability $\frac{1}{2}$ and b with probability $\frac{1}{2}$. Therefore, while the i.i.d. case allows us to predict reliably, in the dependent case there is in general no reliable predictor for the next symbol. In Blackwell and Dubins (1962), however, Blackwell and Dubin show that under certain conditions predictions of two measures merge asymptotically almost surely.

6. Conclusion

Many psychological theories see learning from data, whether sensory or linguistic, as a central function of the brain. Such learning faces great practical difficulties—the space of possible structures is very large and difficult to search, the computational power of the brain is limited, and the amount of available data may also be limited. But it is not clear under what circumstances such learning is possible even with unlimited data and computational resources. Here we have shown that, under surprisingly general conditions, some positive results about identification in the limit in such contexts can be established.

Using an infinite sequence of elements (or a finite sequence of large enough but unknown length) from a set of natural numbers, algorithms are exhibited that identify in the limit the probability distribution associated with this set. This happens in two cases. (i) The underlying set is countable and the target distribution is a probability mass function (i.i.d. measure) in a c.e. or co-c.e. set of computable probability mass functions. The elements of the sequence are drawn i.i.d. according to this probability (Theorem 1). This result is extended to computable ergodic Markov chains (Corollary 1). (ii) The underlying set is countable and the infinite sequence is possibly dependent and is typical for a computable measure in a c.e. or co-c.e. set of computable measures (Theorem 2).

In the i.i.d. case and the ergodic Markov chain case the target is identified in the limit *almost surely*, and in the dependent case the target computable measure is identified in the limit *surely*—however it is not unique but one out of a set of satisfactory computable measures. In the i.i.d. case and Markov case we use the strong law of large numbers. For the dependent case we use typicality according to the theory developed by Martin-Löf in Martin-Löf (1966) which is embedded in the theory of Kolmogorov complexity.

In both the i.i.d., the Markovian, and the dependent settings, eventually we guess an index of the target (or one target out of some possible targets in the measure case) and stick to this guess forever. This last guess is correct. However, we do not know when the guess becomes permanent. We use only a finite unknown-length initial segment of the data sequence. The target for which the guess is correct is described by an appropriate Turing machine computing the probability mass function, Markov chain, or measure, respectively.

These results concerning algorithms for identification in the limit consider what one might term the "outer limits" of what is learnable, by abstracting away from computational restrictions and a finite amount of data available to human learners. Nonetheless, such general results may be informative when attempting to understand what is learnable in more restricted settings. Most straightforwardly, that which is not learnable in the unrestricted case will, *a fortiori*, not be learnable when computational or data restrictions are added. It is also possible that some of the proof techniques used in the present context can be adapted to analyze more restricted, and hence more cognitively realistic, settings.

Acknowledgments

We thank Laurent Bienvenu for pointing out an error in an earlier version and elucidating comments. Drafts of this paper proceeded since 2012 in various states of correctness through arXiv:1208.5003 to arXiv:1311.7385. The second author was supported by ERC grant 295917-RATIONALITY, the ESRC Network for Integrated Behavioral Science (Grant Number ES/K002201/1), the Leverhulme Trust (Grant Number RP2012-V-022), and Research Councils UK Grant EP/K039830/1.

Appendix A. Computability

We can interpret a pair of integers such as (a, b) as rational a/b. A real function f with rational argument is *lower semicomputable* if it is defined by a rational-valued computable function $\phi(x, k)$ with x a rational number and k a nonnegative integer such that $\phi(x, k+1) \geq \phi(x, k)$ for every k and $\lim_{k\to\infty} \phi(x, k) = f(x)$. This means that f can be computably approximated arbitrary closely from below (see Li & Vitányi, 2008, p. 35). A function f is *upper semicomputable* if $-f$ is semicomputable from below. If a real function is both lower semicomputable and upper semicomputable then it is *computable*. A function $f : \mathcal{N} \to \mathcal{R}^+$ is a *probability mass function* if $\sum_x f(x) = 1$. It is customary to write $p(x)$ for $f(x)$ if the function involved is a probability mass function.

A set $A \subseteq \mathcal{N}$ is *computable enumerable* (c.e.) when we can compute the enumeration $a_1, a_2, \ldots$ with $a_i \in A$ $(i \geq 1)$. A c.e. set is also called recursively enumerable (r.e.). A *co-c.e.* set $B \subseteq \mathcal{N}$ is a set whose complement $\mathcal{N} \setminus B$ is c.e. (A set is c.e. iff it is at level Σ_1^0 of the arithmetic hierarchy and it is co-c.e. iff it is at level Π_1^0.) If a set is both c.e. and co-c.e. then it is *computable*. A *halting* algorithm for a *computable function* $f : \mathcal{N} \to \mathcal{R}$ is an algorithm which given an argument x and any rational $\epsilon > 0$ computes a total computable rational function $\hat{f} : \mathcal{N} \times \mathcal{Q} \to \mathcal{Q}$ such that $|f(x) - \hat{f}(x, \epsilon)| \leq \epsilon$.

Example 4. We give an example of the relation between co-c.e. and identification in the limit. Consider a c.e. set A of objects and the co-c.e. set B such that $\mathcal{N} \setminus B = A$. We call the members of B the good objects and the members of A the bad objects. We do not know in what order the bad objects are enumerated or repeated; however we do know that the remaining items are the good objects. These good objects with possible repetitions form the enumeration B. It takes unknown time to enumerate an initial segment of B, but we are sure this happens eventually. Hence to identify the kth element in the enumeration B requires identification of the first $1, \ldots, k-1$ elements. This constitutes identification in the limit. ◊

Example 5. It is known that the overwhelming majority of real numbers are not computable. If a real number a is lower semicomputable but not computable, then we can computably find nonnegative integers $a_1, a_2, \ldots$ and $b_1, b_2, \ldots$ such that $a_n/b_n \leq$

a_{n+1}/b_{n+1} and $\lim_{n\to\infty} a_n/b_n = a$. If a is the probability of success in a trial then this gives an example of a lower semicomputable probability mass function which is not computable. ◊

Suppose we are concerned with all and only computable probability mass functions. There are countably many since there are only countably many computable functions. But can we computably enumerate them?

Lemma 1. (i) *Let $L \subseteq \mathcal{N}$ and infinite. The computable positive probability mass functions on L are not c.e.*
(ii) *Let $L \subseteq \mathcal{N}$ with $|L| \geq 2$. The computable positive measures on L are not c.e.*

Proof. (i) Assume to the contrary that the lemma is false and the computable enumeration is $p_1, p_2, \ldots$. Compute a probability mass function p with $p(a_i) \neq p_i(a_i)$ where $a_i \in L$ is the ith element of L as follows. If i is odd then $p(a_i) := p_i(a_i) + p_i(a_i)p_{i+1}(a_{i+1})$ and $p(a_{i+1}) := p_{i+1}(a_{i+1}) - p_i(a_i)p_{i+1}(a_{i+1})$. By construction p is a computable positive probability mass function but different from any p_i in the enumeration $p_1, p_2, \ldots$.

(ii) The set L^* is c.e. Hence the set of cylinders in L^∞ is c.e. Therefore (ii) reduces to (i). •

Remark 1. Every probability mass function is positive on some support $L \neq \varnothing$ and 0 otherwise. Hence Lemma 1 holds for all probability mass functions. ◊

Appendix B. Kolmogorov complexity

We need the theory of Kolmogorov complexity (Li & Vitányi, 2008) (originally in Kolmogorov, 1965 and the prefix version we use here originally in Levin, 1974). A prefix Turing machine is a Turing machine with a one-way read-only input tape with a distinguished tape cell called the *origin*, a finite number of two-way read–write working tapes on which the computation takes place, an auxiliary tape on which the auxiliary string $y \in \{0,1\}^*$ is written, and a one-way write-only output tape. At the start of the computation the input tape is infinitely inscribed from the origin onwards, and the input head is on the origin. The machine operates with a binary alphabet. If the machine halts then the input head has scanned a segment of the input tape from the origin onwards. We call this initial segment the *program*.

By the construction above, for every auxiliary $y \in \{0,1\}^*$, the set of programs is a prefix code: no program is a proper prefix of any other program. Consider a standard enumeration of all prefix Turing machines

$$T_1, T_2, \ldots.$$

Let U denote a prefix Turing machine such that for every $z, y \in \{0,1\}^*$ and $i \geq 1$ we have $U(i,z,y) = T_i(z,y)$. That is, for each finite binary program z, auxiliary y, and machine index $i \geq 1$, we have that U's execution on inputs i and z, y results in the same output as that obtained by executing T_i on input z, y. We call such a U a *universal* prefix Turing machine.

However, there are more ways a prefix Turing machine can simulate other prefix Turing machines. For example, let U' be such that $U'(i, zz, y) = T_i(z,y)$ for all i and z, y, and $U'(p) = 0$ for $p \neq i, zz, y$ for some i, z, y. Then U' is universal also. To distinguish machines like U with nonredundant input from other universal machines, Kolmogorov (1965) called them *optimal*.

Fix an optimal machine, say U. Define the conditional *prefix Kolmogorov complexity* $K(x|y)$ for all $x, y \in \{0,1\}^*$ by $K(x|y) = \min_p\{|p| : p \in \{0,1\}^* \text{and } U(p,y) = x\}$. (Here U has two arguments rather than three. We consider the first argument to encode the first two arguments of the previous three.) For the same U, define the *time-bounded conditional prefix Kolmogorov complexity* $K^t(x|y) = \min_p\{|p| : p \in \{0,1\}^* \text{and } U(p,y) = x \text{ in t steps}\}$. To obtain the unconditional versions of the prefix Kolmogorov complexities set $y = \epsilon$ where ϵ is the *empty word* (the word with no letters). It can be shown that $K(x|y)$ is uncomputable (Kolmogorov, 1965). Clearly $K^t(x|y)$ is computable if $t < \infty$. Moreover, $K^{t'}(x|y) \leq K^t(x|y)$ for every $t' \geq t$, and $\lim_{t\to\infty} K^t(x|y) = K(x|y)$.

Appendix C. Measures and computability

Let $L \subseteq \mathcal{N}$. Given a finite sequence $x = x_1, x_2, \ldots, x_n$ of elements of L, we consider the set of infinite sequences starting with x. The set of all such sequences is written as Γ_x, the *cylinder* of x. We associate a probability $\mu(\Gamma_x)$ with the event that an element of Γ_x occurs. Here we simplify the notation $\mu(\Gamma_x)$ and write $\mu(x)$. The transitive closure of the intersection, complement, and countable union of cylinders gives a set of subsets of L^∞. The probabilities associated with these subsets are derived from the probabilities of the cylinders in standard ways (Kolmogorov, 1933). A *measure* μ satisfies the following equalities:

$$\mu(\epsilon) = 1 \tag{C.1}$$
$$\mu(x) = \sum_{a\in L} \mu(xa).$$

Let $x_1, x_2, \ldots$ be an infinite sequence of elements of L. The sequence is typical for a computable measure μ if it passes all computable sequential tests (known and unknown alike) for randomness with respect to μ. These tests are formalized by Martin-Löf (1966). One of the highlights of the theory of Martin-Löf is that the sequence passes all these tests iff it passes a single computable universal test, Li and Vitányi (2008, Corollary 4.5.2 on p 315), see also Martin-Löf (1966).

Definition 4. Let $x_1, x_2, \ldots$ be an infinite sequence of elements of $L \subseteq \mathcal{N}$. The sequence is *typical* or *random* for a computable measure μ iff

$$\sup_n \left\{ \log \frac{1}{\mu(x_1 \ldots x_n)} - K(x_1 \ldots x_n) \right\} < \infty. \tag{C.2}$$

The set of infinite sequences that are typical with respect to a measure μ have μ-measure one. The theory and properties of such sequences for computable measures are extensively treated in Li and Vitányi (2008, Chapter 4). There the term $K(x_1 \ldots x_n)$ in (C.2) is given as $K(x_1 \ldots x_n|\mu)$. However, since μ is computable we have $K(\mu) < \infty$ and therefore $K(x_1 \ldots x_n|\mu) \leq K(x_1 \ldots x_n) + O(1)$.

Example 6. Let k be a positive integer and fix an $a \in \{1, \ldots, k\}$. Define measure μ_k by $\mu_k(\epsilon) = 1$ and $\mu_k(x_1 \ldots x_n) = 1/k$ for $n \geq 1$ and $x_i = a$ for every $1 \leq i \leq n$, and $\mu_k(x_1 \ldots x_n) = (1 - 1/k)/(k^n - 1)$ otherwise. Then $K(a \ldots a)$ (a sequence of n elements a) equals $K(a, n) + O(1) = O(\log n + \log k)$. (A sequence of n elements a is described by n in $O(\log n)$ bits and a in $O(\log k)$ bits.) By (C.2) we have $\sup_{n\in\mathcal{N}}\{\log 1/\mu_k(a \ldots a) - K(a \ldots a)\} < \infty$. Therefore the infinite sequence $a, a, \ldots$ is typical for every μ_k. However, the infinite sequence $y_1, y_2, \ldots$ is not typical for μ_k with $y_i \in \{1, \ldots, k\}$ $(1 \leq i \leq k)$ and $y_i \neq y_{i+1}$ for some i. Namely, $\sup_{n\in\mathcal{N}}\{\log 1/\mu_k(y_1 y_2 \ldots y_n) - K(y_1 y_2 \ldots y_n)\} = \infty$. ◊

Since k can be any positive integer, the example shows that an infinite sequence of data can be typical for more than one measure. Hence our task is not to identify a single computable measure according to which the data sequence was generated as a typical sequence, but to identify a computable measure that *could* have generated the data sequence as a typical sequence.

Appendix D. Proofs of the theorems

Proof of Theorem 1 (*i.i.d. Computable Probability Identification*)**.** Let $L \subseteq \mathcal{N}$, and $X_1, X_2, \ldots$ be a sequence of mutually independent random variables, each of which is a copy of a single random variable X with probability mass function $P(X = a) = p(a)$ for $a \in L$. Without loss of generality $p(a) > 0$ for all $a \in L$. Let $\#a(x_1, x_2, \ldots, x_n)$ denote the number of times $x_i = a$ $(1 \leq i \leq n)$ for some fixed $a \in L$.

Claim 1. *If the outcomes of the random variables $X_1, X_2, \ldots$ are $x_1, x_2, \ldots$, then almost surely for all $a \in L$ we have*

$$\lim_{n\to\infty} \left(p(a) - \frac{\#a(x_1, x_2, \ldots, x_n)}{n} \right) = 0. \tag{D.1}$$

Proof. The strong law of large numbers (originally in Kolmogorov, 1930, see also Cantelli, 1917 and Kolmogorov, 1933) states that if we perform the same experiment a large number of times, then almost surely the number of successes divided by the number of trials goes to the expected value, provided the mean exists, see the theorem on top of page 260 in Feller (1968). To determine the probability of an $a \in L$ we consider the random variables X_a with just two outcomes $\{a, \bar{a}\}$. This X_a is a Bernoulli process $(q_a, 1 - q_a)$ where $q_a = p(a)$ is the probability of a and $1 - q_a = \sum_{b\in L\setminus\{a\}} p(b)$ is the probability of $\bar{a}$. If we set $\bar{a} = \min\,(L \setminus \{a\})$, then the mean μ_a of X_a is

$$\mu_a = aq_a + \bar{a}(1 - q_a) \leq \max\{a, \bar{a}\} < \infty.$$

Thus, every $a \in L$ is associated with a random variable X_a with a finite mean. Therefore, $(1/n)\sum_{i=1}^{n}(X_a)_i$ converges almost surely to q_a as $n \to \infty$. The claim follows. •

Let $\mathcal{A}$ be a list of a c.e. or co-c.e. set of halting algorithms for the computable probability mass functions. If $q \in \mathcal{A}$ and $q = p$ then for every $\epsilon > 0$ and $a \in L$ holds $p(a) - q(a) \leq \epsilon$. By Claim 1, almost surely

$$\lim_{n\to\infty} \max_{a\in L} \left(q(a) - \frac{\#a(x_1, x_2, \ldots, x_n)}{n} \right) = 0. \tag{D.2}$$

If $q \in \mathcal{A}$ and $q \neq p$ then there is an $a \in L$ and a constant $\delta > 0$ such that $|p(a) - q(a)| > \delta$. Again by Claim 1, almost surely

$$\lim_{n\to\infty} \max_{a\in L} \left| q(a) - \frac{\#a(x_1, x_2, \ldots, x_n)}{n} \right| > \delta. \tag{D.3}$$

In the proof (Feller, 1968, p. 204) of the strong law of large numbers it is shown that if we draw $x_1, x_2, \ldots$ i.i.d. from a set $L \subseteq \mathcal{N}$ according to a probability mass function p then almost surely the size of the fluctuations in going to the limit (D.2) satisfies $|np(a) - \#a(x_1, x_2, \ldots, x_n)|/\sqrt{np(a)p(\bar{a})} < \sqrt{2\lambda \lg n}$ for every $\lambda > 1$ and n is large enough, for all $a \in L$. Here lg denotes the natural logarithm. Since $p(a)p(\bar{a}) \leq \frac{1}{4}$ and with $\lambda = \sqrt{2}$ it suffices that $|p(a) - \#a(x_1, x_2, \ldots, x_n)/n| < \sqrt{(\lg n)/n}$ for all but finitely many n.

Let $q \in \mathcal{A}$. For $q \neq p$ there is an $a \in L$ such that by (D.3) and the fluctuations in going to that limit we have $|q(a) - \#a(x_1, x_2, \ldots, x_n)/n| > \delta - \sqrt{(\lg n)/n}$ for all but finitely many n. Since $\delta > 0$ is constant, we have $2\sqrt{(\lg n)/n} < \delta$ for all but finitely many n. Hence $|q(a) - \#a(x_1, x_2, \ldots, x_n)/n| > \sqrt{(\lg n)/n}$ for all but finitely many n.

Let $\mathcal{A} = q_1, q_2, \ldots$ and $p = q_k$ with k least. We give an algorithm with as output a sequence of indexes $i_1, i_2, \ldots$ such that all but finitely many indexes are k. If $L = \{a_1, a_2, \ldots\}$ is infinite then the algorithm will only use a finite subset of it. Hence we need to define this finite subset and show that the remaining elements can be ignored. Let $A_n = \{a \in L : \#a(x_1, x_2, \ldots, x_n) > 0\}$. In case $a \in L$ but $a \notin A_n$ we still have $|q_k(a) - \#a(x_1, x_2, \ldots, x_n)/n| \leq \sqrt{(\lg n)/n}$ for all but finitely many n.

Now define the following sets. For each $q_i \in \mathcal{A}$ the set $B_{i,n} = \{a_1, \ldots, a_m\}$ with m least such that $\sum_{j=m+1}^{\infty} q_i(a_j) = 1 - \sum_{j=1}^{m} q_i(a_j) < \sqrt{1/n}$. Therefore, if $a \in L \setminus B_{i,n}$ then $q_i(a) < \sqrt{1/n}$. In contrast to the infinity of L the sets A_n and $B_{i,n}$ are finite for all n and i.

Define $L_{i,n} = A_n \bigcup B_{i,n}$. Since $L_{i,n} \subseteq L$ we have for every $a \in L_{i,n}$ that $|q_k(a) - \#a(x_1, x_2, \ldots, x_n)/n| \leq \sqrt{(\lg n)/n}$ for all but finitely many n. However, for $q_i \neq q_k$ there is an $a \in L_{i,n}$ but no $a \in L \setminus L_{i,n}$ such that $|q_i(a) - \#a(x_1, x_2, \ldots, x_n)/n| > \sqrt{(\lg n)/n}$ for all but finitely many n. This leads to the following algorithm with I the set of indexes of the elements in $\mathcal{A}$:

for $n := 1, 2, \ldots$
  $I := \emptyset$; **for** $i := 1, 2, \ldots, n$
    if $\max_{a\in L_{i,n}} |q_i(a) - \#a(x_1, x_2, \ldots, x_n)/n| < \sqrt{(\lg n)/n}$
    then $I := I \bigcup \{i\}$;
    $i_n := \min I$.

With probability 1 for every $i < k$ for all but finitely many n we have $i \notin I$ while $k \in I$ for all but finitely many n. (Note that for every $n = 1, 2, \ldots$ the main term in the above algorithm is computable even if L is infinite.) The theorem is proven. •

Proof of Theorem 2 (*Computable Measure Identification*)**.** For the Kolmogorov complexity notions see Appendix B. For the theory of computable measures, see Appendix C. In particular we use the criterion of Definition 4 in Appendix C to show that an infinite sequence is typical in Martin-Löf's sense. The given data sequence $x_1, x_2, \ldots$ is by assumption typical for some computable measure μ in a c.e. or co-c.e. set of computable measures and hence satisfies (C.2) with respect to μ. We stress that the data sequence is possibly typical for different computable measures. Therefore we cannot speak of the single *true* computable measure, but only of *a* computable measure for which the data is typical.

Let $\mathcal{B}$ be an enumeration of halting algorithms for a c.e. or co-c.e. set of computable measures such that each element occurs infinitely many times in the list. If the enumeration is such that each element occurs only finitely many times, then the enumeration can be changed into one where each element occurs infinitely many times. For instance, by repeating the first element after every position in the original enumeration, repeating the second element in the original enumeration after every second position in the resulting enumeration, and so on.

Claim 2. *There is an algorithm with as input an enumeration $\mathcal{B} = \mu_1, \mu_2, \ldots$ and as output a sequence of indexes $i_1, i_2, \ldots$. For every large enough n we have $i_n = k$ with μ_k a computable measure for which the data sequence is typical.*

Proof. Define for μ in $\mathcal{B}$

$$\sigma(j) = \log 1/\mu(x_1 \ldots x_j) - K(x_1 \ldots x_j).$$

Since K is upper semicomputable and μ is computable, the function $\sigma(j)$ is lower semicomputable for each j. Define the nth value in the lower semicomputation of $\sigma(j)$ as $\sigma^n(j)$. By (C.2), the data sequence $x_1, x_2, \ldots$ is typical for μ if $\sup_{j\geq 1} \sigma(j) = \sigma < \infty$. In this case, since μ is lower semicomputable, $\max_{1\leq j\leq n} \sigma^n(j) \leq \sigma$ for all n. In contrast, the data sequence is not typical for μ if $\sigma(n) \to \infty$ with $n \to \infty$ implying $\sigma^n(n) \to \infty$ with $n \to \infty$.

By assumption there exists a measure in $\mathcal{B}$ for which the data sequence is typical. Let μ_h be such a measure Since halting algorithms for μ_h occur infinitely often in the enumeration $\mathcal{B}$ there is a halting algorithm $\mu_{h'}$ in the enumeration $\mathcal{B}$ with $\sigma_{h'} = \sigma_h$ and $\sigma_h < h'$. Therefore, there exists a measure μ_k in $\mathcal{B}$ for which the data sequence $x_1, x_2, \ldots$ is typical and $\sigma_k < k$ with k least. The algorithm to determine k is as follows.

for $n := 1, 2, \ldots$
  if $i \leq n$ is least such that $\max_{1 \leq j \leq n} \sigma_i^n(j) < i$
  then output $i_n = i$ **else** output $i_n = 1$.

Eventually $\max_{1 \leq j \leq n} \sigma_k^n(j) < k$ for large enough n, and k is the least index of elements in $\mathcal{B}$ for which this holds. Hence there exists an n_0 such that $i_n = k$ for all $n \geq n_0$. •

For large enough n we have by Claim 2 a test such that we can identify in the limit an index of a measure in $\mathcal{B}$ for which the provided data sequence is typical. Hence there is an n_0 such that $i_n = k$ for all $n \geq n_0$. We do not care what $i_1, \ldots, i_{n-1}$ are. This proves the theorem. •

Appendix E. Genesis of the result

At the request of a referee we give a brief account of the genesis of the result. In version arXiv:1208.5003 we assumed that we were dealing with all computable probabilities and the necessary extensions to measures. The first part of the technical results dealt with i.i.d. drawing and ergodic Markov chains. Here a main ingredient was to appeal to the known result that computable semiprobability mass functions (those summing to 1 or less than 1) are computably enumerable in a linear list. By some tricks we sought to computably extract the probabilities proper from among them and use the Law of Large Numbers. For the more difficult dependent case we resorted to measures. Here we used a known result that the computable semimeasures (where the equality signs in the measure conditions are replaced by inequality $\leq$ signs) are computably enumerable as well in a linear list. Again we sought to computably extract the measures proper from this list and use a (known) criterion that says that the measures for which the provided infinite sequence of examples is random (typical) keeps a certain quantity finite. The proof in arXiv:1208.5003 entailed to separate the finite sequences of this quantity from the infinite ones. This took a long time and great effort. Subsequently in Bienvenu et al. (2014) it was shown that the approach of arXiv:1208.5003 was in error: they showed by a very technical argument that identification of computable probabilities and computable measures by infinite sequences of examples was impossible. Extensive email contact with one of the authors, Laurent Bienvenu, showed that the essential point was the extraction of probabilities and measures from the above computable enumerations of all computable semiprobabilities and computable semimeasures. It turned out that we required computable enumerations or co-computable enumerations of computable probabilities and computable measures at the outset. This was done in arXiv:1311.7385. That is, the identification does not hold for all computable probabilities and computable measures as in the too large claims of arXiv:1208.5003 but for the subclass of computable enumerations or co-computable enumerations of them. Furthermore the very difficult argument separating bounded infinite sequences from unbounded ones (in the dependent case) was replaced by a simple one reminiscent of the h-index in citation science. Namely, a bounded infinite sequence has a(n unknown) bound. But if the measures involved are enumerated then eventually the index of one (there are infinitely many of them) for which the bound is relevant will pass this bound.

References

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, *9*(1), 147–169.

Angluin, D. (1988). *Identifying languages from stochastic examples. Technical Report*. New Haven, Conn., USA: Yale University, Dept. of Computer Science.

Baker, C. L., & McCarthy, J. J. (1981). *The logical problem of language acquisition*. Cambridge, MA: MIT Press.

Barron, A. R., & Cover, T. M. (1991). Minimum complexity density estimation. *IEEE Transactions on Information Theory*, *4*, 1034–1054.

Bienvenu, L., Monin, B., & Shen, A. (2014). Algorithmic identification of probabilities is hard. In *Springer lecture notes in artificial intelligence: Vol. 8776. Proc. algorithmic learning theory* (pp. 85–95).

Blackwell, D., & Dubins, L. (1962). Merging of opinions with increasing information. *The Annals of Mathematical Statistics*, *33*, 882–886.

Bowerman, M. (1988). The 'No Negative Evidence' problem: How do children avoid constructing an overly general grammar? In J. Hawkins (Ed.), *Explaining language universals* (pp. 73–101). Oxford: Blackwell.

Cantelli, F. P. (1917). Sulla probabilitá come limite della frequenza. *Rendiconti della R. Accademia dei Lincei, Classe di Scienze Fisische Matematiche e Naturale, Serie 5A*, *26*, 39–45.

Charniak, E. (1996). *Statistical language learning*. Cambridge, MA: MIT Press.

Chater, N., Clark, A., Goldsmith, J., & Perfors, A. (2015). *Empiricist Approaches to Language Learning*. Oxford, UK: Oxford University Press.

Chater, N., & Oaksford, M. (2008). *The probabilistic mind: Prospects for Bayesian cognitive science*. Oxford: Oxford University Press.

Chater, N., Tenenbaum, J. B., & Yuille, A. (2006). Probabilistic models of cognition: Conceptual foundations. *Trends in Cognitive Sciences*, *10*, 287–291.

Chater, N., & Vitányi, P. M. B. (2007). Ideal learning of natural language: Positive results about learning from positive evidence. *Journal of Mathematical Psychology*, *51*, 135–163.

Chomsky, N. (1980). Rules and representations. *Behavioral and Brain Sciences*, *3*, 1–15.

Chomsky, N. (1982). *Some concepts and consequences of the theory of government and binding*. Cambridge, MA: MIT Press.

Christiansen, M., & Chater, N. (2016). The now-or-never bottleneck: A fundamental constraint on language. *Behavioral and Brain Sciences*, *39*, e62.

Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, *36*, 181–204.

Clark, A., & Lappin, S. (2010). *Linguistic nativism and the poverty of the stimulus*. Hoboken, NJ: John Wiley and Sons.

Crain, S., & Lillo-Martin, D. C. (1999). *An introduction to linguistic theory and language acquisition*. Malden, MA: Blackwell.

Deutsch, D. (1985). Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. Series A. Mathematical, Physical and Engineering Sciences*, *400*, 97–117.

Duhem, P. (1914–1954). *The aim and structure of physical theory*. Princeton, NJ: Princeton University Press, translated from 2nd edition by P. W. Wiener; originally published as La Théorie Physique: Son Objet et sa Structure (Paris: Marcel Riviera & Cie).

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.

Feller, W. (1968). *An introduction to probability theory and its applications, Vol. 1* (3rd ed.). New York: Wiley.

Gold, E. M. (1965). Limiting recursion. *Journal of Symbolic Logic*, *30*, 28–48.

Gold, E. M. (1967). Language identification in the limit. *Information and Control*, *10*, 447–474.

Gopnik, A., Meltzoff, A. N., & Kuhl, P. K. (1999). *The scientist in the crib: Minds, brains, and how children learn*. New York: William Morrow & Co..

Haber, R. N. (1983). The impending demise of the icon: A critique of the concept of iconic storage in visual information processing. *Behavioral and Brain Sciences*, *6*, 1–54.

Hahn, U., & Oaksford, M. (2008). Inference from absence in language and thought. In M. Oaksford, & N. Chater (Eds.), *The probabilistic mind: Prospects for Bayesian cognitive science* (pp. 121–142). Oxford: Oxford University Press.

Hollerman, J. R., & Schultz, W. (1998). Dopamine neurons report an error in the temporal prediction of reward during learning. *Nature Neuroscience*, *1*, 304–309.

Hornstein, N., & Lightfoot, D. (1981). *Explanation in linguistics. The logical problem of language acquisition*. London: Longman.

Hsu, A., Chater, N., & Vitányi, P. M. B. (2011). The probabilistic analysis of language acquisition: Theoretical, computational, and experimental analysis. *Cognition*, *120*, 380–390.

Hsu, A. S., Horng, A., Griffiths, T. L., & Chater, N. (2016). When absence of evidence is evidence of absence: Rational inferences from absent data. *Cognitive Science*, *40*, 1–13.

Jain, S., Osherson, D. N., Royer, J. S., & Sharma, A. (1999). *Systems that learn* (2nd ed.). Cambridge, MA: MIT Press.

Joshi, A. K., & Schabes, Y. (1997). Tree-adjoining grammars. In *Handbook of formal languages* (pp. 69–123). Berlin: Springer.

Kilner, J. M., Friston, K. J., & Frith, C. D. (2007). Predictive coding: an account of the mirror neuron system. *Cognitive Processing*, *8*, 159–166.

Knill, D. C., & Pouget, A. (2004). The Bayesian brain: the role of uncertainty in neural coding and computation. *Trends in Neurosciences*, *27*, 712–719.

Kolmogorov, A. N. (1930). Sur la loi forte des grandes nombres. *Comptes Rendus de l'Academie des Sciences, Serie Generale, la Vie des Sciences*, *191*, 910–912.

Kolmogorov, A. N. (1933). *Grundbegriffe der wahrscheinlichkeitsrechnung*. Berlin, Berlin: Springer-Verlag.

Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. *Problems of Information Transmission*, *1*, 1–7.

Lange, K. (2005). *Applied probability*. Springer, (Corrected 2nd printing).

Levin, L. A. (1974). Laws of information conservation (non-growth) and aspects of the foundation of probability theory. *Problems of Information Transmission*, *10*, 206–210.

Li, S. Z. (2012). *Markov random field modeling in computer vision*. Berlin: Springer.

Li, M., & Vitányi, P. M. B. (2008). *An introduction to Kolmogorov complexity and its applications* (3rd ed.). New York: Springer-Verlag.
Manning, C., & Klein, D. (2003). Natural language parsing. In *Advances in neural information processing systems 15: proceedings of the 2002 conference, Vol. 15*. Cambridge, MA: MIT Press.
Martin-Löf, P. (1966). The definition of random sequences. *Information and Control*, *9*, 602–619.
Oaksford, M., & Chater, N. (2007). *Bayesian rationality: The probabilistic approach to human reasoning*. Oxford: Oxford University Press.
Pearl, J. (2014). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Burlington, MA: Morgan Kaufmann.
Pickering, M. J., & Garrod, S. (2013). An integrated theory of language production and comprehension. *Behavioral and Brain Sciences*, *36*, 329–347.
Pinker, S. (1979). Formal models of language learning. *Cognition*, *7*, 217–283.
Popper, K. R. (1959). *The logic of scientific discovery*. London: Hutchinson.
Pouget, A., Beck, J. M., Ma, W. J., & Latham, P. E. (2013). Probabilistic brains: knowns and unknowns. *Nature Neuroscience*, *16*, 1170–1178.
Pullum, G. K., & Scholz, B. C. (2002). Empirical assessment of stimulus poverty arguments. *The Linguistic Review*, *18*, 9–50.
Quine, W. V. O. (1951). *Two dogmas of empiricism* (2nd ed.) (pp. 20–46). Cambridge, MA: Harvard University Press, Reprinted in From a Logical Point of View.
Reber, A. S. (1989). Implicit learning and tacit knowledge. *Journal of Experimental Psychology: General*, *118*(3), 219.
Rescorla, M. (2015). The computational theory of mind. In E. N. Zalta (Ed.), *The stanford encyclopedia of philosophy* (Winter 2015 ed.). URL http://plato.stanford.edu/archives/win2015/entries/computational-mind/.
Rescorla, R. A., & Wagner, A. R. (1972). A theory of Pavlovan conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In A. H. Black, & W. F. Prokasy (Eds.), *Classical conditioning II: current theory and research* (pp. 64–99). New York: Appleton-Century.
Saffran, J. R., Aslin, R. N., & Newport, E. L. (1996). Statistical learning by 8-month-old infants. *Science*, *274*, 1926–1928.
Shannon, C. E. (1951). Prediction and entropy of printed English. *Bell System Technical Journal*, *30*, 50–64.
Shepard, R. N. (1984). Ecological constraints on internal representation: resonant kinematics of perceiving, imagining, thinking, and dreaming. *Psychological Review*, *91*, 417–447.
Shepard, R. N., Hovland, C. I., & Jenkins, H. M. (1961). *Psychological monographs: general and applied*: *Vol. 75*. *Learning and memorization of classifications* (pp. 1–42).
Solomonoff, R. J. (1964). A formal theory of inductive inference, part 1 and part 2. *Information and Control*, *7*(1–22), 224–254.
Steedman, M. (2000). *The syntactic process*. Cambridge: MIT Press.
Tenenbaum, J. B., Kemp, C., Griffiths, T. L., & Goodman, N. D. (2011). How to grow a mind: Statistics, structure, and abstraction. *Science*, *331*, 1279–1285.
Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, *2*, 230–265.
Yuille, A., & Kersten, D. (2006). Vision as Bayesian inference: analysis by synthesis? *Trends in Cognitive Sciences*, *10*, 301–308.
Zeugmann, T., & Zilles, S. (2008). Learning recursive functions: a survey. *Theoretical Computer Science*, *397*, 4–56.