Available online at www.sciencedirect.com
ScienceDirect
Journal of Mathematical Psychology 51 (2007) 135–163
Journal of Mathematical Psychology
www.elsevier.com/locate/jmp

'Ideal learning' of natural language: Positive results about learning from positive evidence

Nick Chater[a,*], Paul Vitányi[b]

[a]Department of Psychology, University College, London WC1E 6BT, UK
[b]Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands

Received 25 November 2005; received in revised form 5 September 2006
Available online 26 December 2006

Abstract

Gold's [1967. Language identification in the limit. *Information and Control*, *16*, 447–474] celebrated work on learning in the limit has been taken, by many cognitive scientists, to have powerful negative implications for the learnability of language from positive data (i.e., from mere exposure to linguistic input). This provides one, of several, lines of argument that language acquisition must draw on other sources of information, including innate constraints on learning. We consider an 'ideal learner' that applies a Simplicity Principle to the problem of language acquisition. The Simplicity Principle chooses the hypothesis that provides the briefest representation of the available data—here, the data are the linguistic input to the child. The Simplicity Principle allows learning from positive evidence alone, given quite weak assumptions, in apparent contrast to results on language learnability in the limit (e.g., Gold, 1967). These results provide a framework for reconsidering the learnability of various aspects of natural language from positive evidence, which has been at the center of theoretical debate in research on language acquisition and linguistics.
© 2006 Elsevier Inc. All rights reserved.

Keywords: Learnability; Language acquisition; Algorithmic complexity; Kolmogorov; Identification in the limit; Formal languages

1. Introduction

Language acquisition involves the rapid mastery of linguistic structure of astonishing complexity based on an input that appears noisy and partial. How can such an impoverished stimulus support such impressive learning? One influential line of argument is that it cannot—this "poverty of the stimulus" argument (Chomsky, 1980) is typically used to argue that language acquisition is guided by innate knowledge of language, often termed "universal grammar," that the child brings to bear on the learning problem (e.g., Chomsky, 1965, 1980; Hoekstra & Kooij, 1988). This type of argument for universal grammar is of central importance for the study of human language and language acquisition (e.g., Crain & Lillo-Martin, 1999; Hornstein & Lightfoot, 1981).

How can the poverty of the stimulus argument be assessed? At an abstract level, a natural approach is to attempt to somehow define an "ideal" language learner, which lacks universal grammar, but that can make the best use of the linguistic evidence that the child is given. If it were possible to show that this ideal learner is unable to learn language from the specific linguistic data available to the child, then we might reasonably conclude that some innate information must be available. Indeed, a second step, although not one we will consider in this paper, would be to attempt to prove that the ideal language learner, when provided with some appropriate innate information, is then able to learn language successfully from data of the sort available to the child.

Clearly, the task of constructing such an ideal language learner is a formidable one. We might reasonably suspect that the project of finding an optimal way of learning language is inherently open-ended; and our present understanding both of the mechanisms of human learning, and the computational and mathematical theory of learning, is sufficiently undeveloped that it is clear that the project is, in full generality, well beyond the scope of current research.

*Corresponding author. Fax: +44 7436 4276.
E-mail address: n.chater@ucl.ac.uk (N. Chater).

0022-2496/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jmp.2006.10.002

How is it, then, that many researchers are already convinced that, whatever form such an ideal learner might take, there is not enough information in the child's language input to support language acquisition, without recourse to universal grammar? Two types of argument have been proposed. The first considers the problem of language acquisition to be in principle problematic. It is argued that there is a *logical* problem of language acquisition—essentially because the child has access only to positive linguistic data (Baker & McCarthy, 1981; Hornstein & Lightfoot, 1981). Despite some controversy, it is now widely assumed that negative linguistic data is not critical in child language acquisition. Children acquire language even though they receive little or no feedback indicating that particular utterances are ungrammatical; and even where they do receive such feedback, they seem unwilling to use it (e.g., Brown & Hanlon, 1970). But learning from positive data alone seems "logically" problematic, because there appears to be no available data to allow the child to recover from overgeneralization. If this line of argument is correct, then whatever ideal learner we might describe, it will inevitably face these logical problems; and hence it will be unable to learn from positive evidence alone.

The second, and closely related, type of argument, focuses on patterns of acquisitions of particular types of linguistic construction and argues that these specific constructions cannot be learned from positive data only (e.g., Baker, 1979; Chomsky, 1980). This type of argument is sometimes labeled Baker's paradox, to which we return below. The nature of this type of argument is necessarily informal—various possible mechanisms for learning the problematic construction of interest are considered, and rejected as unviable.

We shall present mathematical results concerning an ideal learner, with relevance to both issues. In particular, we shall show that, in a specific probabilistic sense, language is learnable, given enough positive data, given only extremely mild computability restrictions on the nature of the language concerned. Thus, the apparent logical problem of language acquisition must be illusory—because a specific ideal learner can demonstrably learn language. Our arguments also, *a fortiori*, address the construction-specific version of the poverty of the stimulus argument. If language as a whole can be learned from positive data, then any specific linguistic construction can be learned from positive data, despite intuitions to the contrary. Indeed, we shall see that there is a general mechanism for such learning—one that has frequently been described, though sometimes dismissed, in discussions of poverty of the stimulus arguments (e.g., Pinker, 1979, 1984). That is, the *absence* of particular linguistic constructions in the positive data can serve as evidence that these structures are not allowed in the language. This point is discussed further below in our discussion of what we call the overgeneralization theorem, which shows how the ideal learner is able to eliminate overgeneral models of the language.

The results presented here should not, though, be viewed as showing that language *is* learnable by children from positive data. This is for two reasons. First, results concerning an ideal learner merely show that the information required to support learning is present *in principle*. It does not, of course, show that the child has the learning machinery required to extract it. Indeed, the ideal learner we consider here is able to make calculations that are known to be uncomputable—and it is typically assumed that the brain is limited to the realm of the computable. Thus an interesting open question for future research concerns learnability results that can be proved with a more restricted ideal learner. The second reason that this work does not show that the child can learn language from positive data is that the results we describe are asymptotic—that is, we allow that the child can have access to as much positive data as required. In practice, though, children learn specific linguistic constructions having heard specific amounts of positive linguistic data, with a specific degree of incompleteness, errorfulness, and so on. The formal results that we describe here do not directly address the question of the speed of learning. Nonetheless, this is typically also true of both logical and construction-specific poverty of the stimulus arguments. Both types of argument typically suggest that, however much positive data is provided, learning will not be successful. These results presented here therefore address these arguments; and raise the question of how to provide specific bounds on the amount of positive data that is required by the learner to learn specific linguistic phenomena.

The formal results in this paper, then, aim to sharpen the discussion of poverty of the stimulus arguments, rather than to resolve the issue one way or the other. According to the results we present, there is enough information in positive linguistic data for language to be learnable, in principle, in a probabilistic sense, given sufficient linguistic data. A challenge for future work on the poverty of the stimulus argument is to sharpen existing arguments, and current formal results, to address the question of what increasingly realistic learners might be able to acquire from increasingly realistic models of the amount and properties of the linguistic input available to the child. In particular, it is interesting to ask whether it is possible to 'scale-down' the methods that we describe here, to explore the question of whether there is sufficient information in the linguistic input available to the child to acquire specific linguistic phenomena that have been viewed as posing particularly difficult problems for the language learner. We hope this work will feed into the current debate in linguistics and psychology concerning the scope and validity of poverty of the stimulus arguments (e.g., Akhtar, Callanan, Pullum, & Scholz, 2004; Fodor & Crowther, 2002; Legate & Yang, 2002; Lidz, Waxman, & Freedman, 2003; Perfors, Tenenbaum, & Regier, 2006; Pullum & Scholz, 2002; Regier & Gahl, 2004; Tomasello, 2004).

The ideal learner that we analyse is based on a *Simplicity Principle*. Roughly, the idea is that the learner postulates

the underlying structure in the linguistic input that provides the simplest, i.e., briefest, description of that linguistic input. We require that the description can actually be used to reconstruct the original linguistic input using some computable process—thus, the goal of the ideal learner is to find the shortest computer program that encodes the linguistic input. The general idea that cognition is a search for simplicity has a long history in psychology (Koffka, 1962/1935; Mach, 1959/1886), and has been widely discussed, in a relatively informal way, in the field of language and language learning (e.g., Fodor & Crain, 1987). We describe a formal theory of inductive reasoning by simplicity, based on the branch of mathematics known as Kolmogorov complexity theory (Li & Vitányi, 1997). Kolmogorov complexity was developed independently by Solomonoff (1964), Kolmogorov (1965) and Chaitin (1969). Solomonoff's primary motivation in developing the theory was to provide a formal model of learning by simplicity. Kolmogorov complexity and derivatives from it have been widely used in mathematics (e.g., Chaitin, 1987), physics (Zurek, 1991), computer science (e.g., Paul, Seiferas, & Simon, 1981), artificial intelligence (Quinlan & Rivest, 1989), and statistics (Rissanen, 1987, 1989; Wallace & Freeman, 1987). This mathematical framework provides a concrete and well-understood specification of what it means to learn by choosing the simplest explanation, and provides a way of precisely defining the Simplicity Principle for cognitive science (Chater, 1996, 1997, 1999; Chater & Vitányi, 2002). Moreover, the Simplicity Principle has been used practically in a wide range of models of language processing and structure (e.g., Brent & Cartwright, 1996; Dowman, 2000; Ellison, 1992; Goldsmith, 2001; Onnis, Roberts & Chater, 2002; Wolff, 1988). This framework will prove to be useful in considering the amount of information available about the language that is inherent in positive evidence alone.

The first substantive section of this paper, *Ideal language learning by simplicity*, outlines the framework for ideal language learning. Roughly, as we have said, the learner finds the shortest "computer program" that can reconstruct the linguistic data that has so far been encountered; it then makes *predictions* about future material based on what that computer program would produce next. The third section, *The prediction theorem and ideal language learning*, presents a remarkable mathematical result, due to Solomonoff (1978), that shows that this learning method is indeed, in a certain sense, ideal. This method learns to make accurate predictions (with high probability) about the language input, given mild computability constraints on the processes generating the linguistic data. The subsequent two sections presents and proves new mathematical results.

In *The ideal learning of grammaticality judgments*, we show how Solomonoff's prediction theorem can be used to show that the ideal learner can, in a probabilistic sense, learn to make arbitrarily good grammaticality judgments. This result is particularly important, given that grammaticality judgments are the primary data of modern linguistic theory, and they are frequently thought to embody information that cannot readily be extracted from corpora of language. Intuitively, the idea is that sentences which are predicted with non-zero probability are judged to be grammatical; sentences that have zero probability are judged to be ungrammatical. Note that this result does not allow for the errorful character of linguistic input—although extensions to this case may be feasible.

In *The ideal learning of language production*, we show that prediction can also allow the ideal language learner to *produce* language that is, with high probability, indistinguishable from the language that it has heard. Intuitively, the idea is that the ability to predict what others might say can be recruited to determine what the speaker should say. Of course, language production is much more than this—in particular, it requires the ability not merely to continue conversations plausibly, but to say things that reflect one's particular beliefs and goals. Nonetheless, the result that an ideal language learner's can continue conversations plausibly is non-trivial. It requires, among other things, the ability to produce language that respects the full range of phonological, grammatical, pragmatic, and other regularities governing natural language.

Finally, in *The poverty of the stimulus reconsidered*, we relate these results to the logical version of the poverty of the stimulus (relating these results to Gold's (1967) results, and later work); and to the construction-specific version (reconsidering Baker's paradox); and we consider open-questions for the approach that we have described. We leave a detailed analysis of the methodological and theoretical implications for the poverty of the stimulus argument, and a whether a Simplicity Principle might explain some aspects of human language acquisition, to future work.

2. Ideal language learning by simplicity

To specify a set-up for learning a language from positive examples, we need to provide (1) the class of linguistic inputs to be learned (the linguistic 'environment'); (2) the class of possible models of the language; (3) a measure of learning performance; and (4) a learning method. Below, the class of linguistic inputs will be those that can be generated by a computable process combined with random noise. Learning performance will be measured both in terms of the ability to predict new input and in relation to judgments of grammaticality. And the learning method will be based on the Simplicity Principle—that the learner should prefer hypotheses which provide the simplest explanations of the linguistic input. We consider each of these points in more detail.

2.1. The class of allowable linguistic inputs

Let us assume that linguistic input corresponds to a potentially infinite sequence of atomic symbols (an

'alphabet'). Without loss of generality, we assume that the stream of sentences that form the input to the learner can be represented as a continuous binary string: a sequence of 0 and 1 s. A simple way of doing this is to associate a fixed binary string with each element of the original alphabet in which the language is expressed (e.g., the standard English alphabet and punctuation symbols for sentences of English). Thus, the sequence of sentences of the input can be converted from its standard representation into a continuous binary sequence.[1]

We assume, further, that the linguistic input is generated by a real computational process. But how, exactly, should we model what counts as a 'real computational process'? A first suggestion would limit possible linguistic inputs to those that can be generated by Turing machines. But this would be over-restrictive, because linguistic input can also be affected by random factors. For example, imagine a speaker reporting successive tosses of a fair coin: "Heads, heads, tails, tails, tails, heads…"and so on. Assuming that the coin is tossed fairly, the corresponding utterance is a random infinite sequence, which cannot be generated by any Turing machine (Li & Vitányi, 1997). Real language input to the child is presumably a mixture of both kinds of factor—deterministic computational processes in the speaker (and perhaps also in other aspects of the environment which the speaker is describing or reacting to), mixed with random influences.[2]

So how can we model this mixture of deterministic and random factors? One natural approach (which turns out to be quite elegant and general) is to assume that language is generated by a *deterministic machine* (concretely, a Turing machine), fed with a *random input* (concretely, the flips of an unbiased coin, which produce a potentially infinite string of binary inputs). As the stream of random input comes in, the machine writes its output as a binary string on a dedicated output tape—this corresponds to the utterances produced by the machine. This set-up needs to reflect a basic fact about utterances—that once they have be said, they cannot then be 'unsaid.' This can be expressed by saying that symbols on the output cannot be deleted. The intuitive picture to have in view is that, as the input grows, the output (the corpus of things 'said') gradually increases, and can never decrease.[3] Let us call a deterministic Turing machine which has this property a *monotone* Turing machine (Li & Vitányi, 1997).[4]

We now have a model for how the linguistic input presented to the learner is generated. It is generated by a mixture of random factors (captured by the random input to the machine); and deterministic factors (captured by an arbitrary monotone Turing machine). The deterministic factors determine which sentences of the language are allowed, and their probabilities—the random factors generate an actual corpus of language, by choosing a particular stream of sentences. The output of the machine is a finite[5] or infinite binary sequence, which can be viewed as encoding the corpus of linguistic material to which the learner is exposed. Because of the random component, many different outputs (corpora) can potentially be generated, and some corpora will be more likely to be generated than others. The probability associated with each output sequence x is the probability that x will be the output of the computational process, when supplied with random binary input. Hence we can associate any given (monotone) computational process, C, with a probability distribution, $\mu_C(x)$, over the set of finite and infinite binary output sequences, x. The fundamental assumption concerning the nature of the linguistic input outlined in this subsection can be summarized as the assumption that the linguistic input is generated by some monotone computable probability distribution $\mu_C(x)$.

But is the definition of monotone computability sufficiently broad? Perhaps language is generated by some cognitive process which is not monotone computable. There is no definitive way to ascertain whether this occurs—a 'monotone computability' thesis must stand as a conjecture, which cannot be verified although it may, at some time in the future, be falsified. There are three points to consider.

The first point is the assumption that the 'deterministic' component is no more powerful than a Turing machine. This seems relatively uncontroversial, because it is in line

[1]The only subtlety here is that the mapping into the binary alphabet should be reversible, meaning that the original alphabetic representation can be uniquely decoded. This can be ensured by, for example, using a *prefix* binary code for the original alphabet and punctuation marks—that is, a code such that no initial portion (i.e., prefix) for any item corresponds to the code for some other item.

[2]No great metaphysical weight needs to be borne by the concept of randomness here. What matters is that many aspects of linguistic input (e.g., those affected by coin tosses, the weather, and 'chance' events of all kinds) will be, from a practical point of view, random for the learner. That is, no underlying pattern can conceivably be found by the learner, whether or not some such pattern ultimately exists. This epistemic notion of randomness is made precise by defining random sequences as sequences that are their own shortest description, leading to the mathematical theory of algorithmic randomness (Li & Vitányi, 1997).

[3]Technically, it is allowed that, at some point, no further output might be produced.

[4]More precisely, the requirement is that the output is a produced by a *monotone* computational process acting on the input. We define a monotone computational process as follows: it reads its input in one direction only (i.e., it cannot 'go back' to look at earlier inputs, although it can store this input in its memory); and it cannot modify this input (the input is 'read-only'). Moreover, the output can be written in one direction only (i.e., once an output is 'written' it cannot be altered); and the output cannot be read (the output is 'write-only'). The output of the machine is defined as the binary sequence on the output tape, if the machine halts (and hence all subsequent inputs are ignored); and the infinite sequence binary sequence on the output tape, if the machine does not halt, but continues producing further outputs indefinitely. See Li and Vitányi (1997, pp. 276–277) for a rigorous description. Thus, as input is added, output cannot be deleted—although it is possible that the machine becomes 'mute'—it produces no more output after a certain point.

[5]The output is finite if the machine produces no more output after a certain point in the infinite binary input sequence. For example, the machine might halt, or go into an infinite loop.

with standard assumptions in cognitive science that cognitive processes are computable (and the Church-Turing thesis, that any computational process can be modeled by a Turing machine). Thus the deterministic component does not seem overly restrictive.

The second point appears more problematic. The assumption that the input to the machine is the result of tossing a fair coin seems extremely restrictive. But, fortunately, this restriction can be weakened without affecting the class of probability distributions that are allowed. Specifically, the assumption can be weakened so that the input is generated by any 'enumerable semi-measure.' Roughly, this is the class of distributions that can be *approximated* in the limit[6] by a computational process (Li & Vitányi, 1997). This is a very broad class of probabilistic inputs indeed and includes all the distributions standardly used in mathematics and probability theory.[7]

The third point concerns whether the sharp separation between deterministic computational processes and purely random factors is unrealistic. For example, on the natural assumption that the computations of the cognitive system are disrupted by internal 'neural' noise, then some of randomness will creep into the workings of the computational process itself. Fortunately, however, incorporating finite amounts of noise into the workings of the computational process itself does not affect the class of probability distributions over linguistic outputs that can be generated. This is because any finite amount of randomness in the internal workings of the machine can be simulated by a deterministic machine that 'absorbs' the relevant amount of randomness from part of its random input.[8]

We shall therefore tentatively adopt the monotone computability conjecture henceforth, and therefore assume that language can be viewed as generated by a fair random binary input into an arbitrary deterministic, monotone computational process.

To get an intuitive sense of how language generation occurs according to the present model, consider a variant of a well-worn example. It is often remarked that a monkey hitting typewriter keys at random would eventually produce the works of Shakespeare. Here, instead, the monkey is hitting the keys, not of a typewriter, but of a programmable computer. So the analogous remark is that a monkey randomly hitting the keys of a programmable computer will eventually write a *computer program* that produces the works of Shakespeare.

The change from random *typing* to random *programming* is of vital importance. In the former case, the probability of a string being generated depends *only* on its length. So the probability of any billion symbol binary string being generated random is the probability of a billion successive coin tosses coming up in some specific way, i.e., $2^{-1000,000,000}$. Thus, all binary strings, whether they encode Shakespeare or are completely random are treated as having precisely the same probability. But this would be of no use as a model of language generation—because in providing the same probability to all outputs, the 'monkey-at-the-typewriter' scenario completely fails to favor grammatical, or relevant, or coherent, utterances over unintelligible gibberish.

By contrast, the 'monkey-at-the-programmable-computer' scenario, allows for such biases to be incorporated into the structure of the computer. These biases can take forms as varied as the possible computer programs. To choose a simple example of relevance to language generation, the programmable computer might, for example, support a 'programming language' in which the rules of a phrase structure grammar can be specified (in this case by the random input). When run, such a program then generates a binary output, encoding a sequence of sentences that are in accordance with those rules. In this case, some binary strings (those that correspond to sequences of sentences which are grammatical according to a phase structure grammar) will be relatively probable; and others (those that do not correspond to such sequences) will have probability 0. So while the monkey *typing* at random is equally likely to produce any random sequence, the monkey *programming* at random (on this particular language generating 'machine') is very much more likely to produce some outputs than others.

The specific probability distribution over binary outputs depends crucially, of course, on the choice of programmable computer, C, into which the random binary input is fed. But if we fix a particular C, we can quantify the probability that a particular output, x, will be generated from random input quite straightforwardly. This probability depends on the number of different 'programs,' y, that produce output x, and on the *length* of each of these programs, $l(y)$.

Focussing on a specific program, y, the probability that it is generated from random input is the probability that first $l(y)$ symbols in the random input correspond precisely to y. There is a probability 1/2 that each particular symbol will correspond to the corresponding symbol of the sequence y, and hence a probability $2^{-l(y)}$ that all $l(y)$ symbols correspond to y. This means that outputs which correspond to *short* programs for the computer, C, are overwhelmingly more probable than outputs for which there are no short programs in C. Now we can derive the

[6]Strictly, approximated in the limit from below.

[7]Provided that these distributions have rational parameters.

[8]This class of outputs of the machine is broader, however, if the internal noise in the system can contribute an *infinite* amount of randomness—more technically, if the internal randomness supplies an infinite number of bits of information. This is because the model presented here only allows a finite amount of randomness to be absorbed from the environment in making any particular output. For example, a computational process which depended on the real valued variable sampled from a probability density function—i.e., where the value of this variable must be known to *infinite* precision in order to assess its computational significance—could not be simulated by the model described here. It is conceptually possible that this might arise—but this assumption is not embodied in any current theoretical and computational model of language processing, to our knowledge.

total probability, $\mu_C(x\ldots)$ that an output beginning with, $x\ldots$, is generated by random input to the computer C. We just need to sum the probabilities of all the inputs y which produce an output $x\ldots$ (i.e., an output that begins with the subsequence x) when run on the C (in symbols, all the y such that $C(y\ldots) = x\ldots$). So the total probability, $\mu_C(x\ldots)$, that $x\ldots$ is generated from random input to C is:[9]

$$\mu_C(x\ldots) = \sum_{y:C(y\ldots)=x\ldots} 2^{-l(y\ldots)}, \tag{1}$$

where $x\ldots$ denote a finite or infinite sequence that begins with the subsequence x, and similarly for $y\ldots$We shall neglect the ellipsis "…" below. Thus, the fundamental assumption of the framework presented here is that the (possibly infinite) corpus of utterances which forms the corpus, x, for the language learner is generated by a probability distribution $\mu_C(x)$, associated with a monotone computational process, C, provided with fair coin flips.[10] We shall call such $\mu_C(x)$ monotone computable probability distributions. We show below that the structure of corpora generated in this way can be learned, to an approximation, from positive evidence alone.

Finally, note that we have skated over some technical complications in the interests of clarity. First, defining probability distributions over infinite sequences requires care, because probabilities typically all tend to zero in the limit. Technically, this requires introducing the notion of a *measure* (we introduce measures, from a non-standard point of view (Li & Vitányi, 1997), which will be useful in subsequent proofs, in Appendix A). Second, the 'probability distributions' we have been discussing do not all obey a standard feature of probability: that the probabilities of outcomes sum to 1. In particular, this arises because some inputs to a monotone machine may produce no well-defined output (the monkey typing into the computer may type a syntactically invalid program, or a program which goes into an infinite loop, or exhibits some other pathology). This means that the sum of the probabilities of the well-defined outputs, s, which we wrote $\mu_C(s)$, may be less than 1. This complication requires generalizing from the conventional measures used in probability theory to the notion of a *semi*-measure (Appendix A). Where measures and semi-measures are drawn on in the results below, the discussion is relegated to other Appendices. Thus, the reader may ignore such complications without substantial loss of comprehension or continuity below.

[9]Eq. (1) is a simplification, because it ignores the issue of double counting two input sequences which both start with a sub-sequence z, and where z alone generates x. See Li and Vitányi (1997) for a rigorous specification, which takes account of this problem. We have ignored this subtlety here and elsewhere below in the interests of clarity.

[10]Or some other enumerable (semicomputable) probability distribution. This is a very broad class of distributions, including all those that are used in statistics (see Li & Vitányi, 1997). It disallows, though, distributions with arbitrary, infinite precision, real parameters, etc. (see footnote 7). These do not, of course, arise in practice in statistics, which inevitably works with finite precision approximations.

2.2. The class of possible models of the language

Many standard formal analyses of the problem of language acquisition provide specific constraints on the types of language that are learnable. From the point of view of Gold's (1967) framework of identification in the limit, for example, the goal of the learner is to identify the language by picking out one of a (typically infinite) set of possible languages: for example, the set of finite state languages, context-free languages, or, from the point of view of Chomsky's (1981) principles and parameters framework, the class of languages defined by the possible values of a finite set of parameters (Gibson & Wexler, 1994). Here, instead the class of languages from which the learner must choose is less constrained; the learner is, in principle, able to entertain all possible processes that might have generated the input. That is, the only restriction is that the model of the linguistic data can be generated by a computable process, supplied, if required, with random input, as described above. Of course, this weak assumption includes all the grammatical formalisms currently used in linguistics and computational linguistics.

2.3. Measuring learning performance

We have defined the class of monotone computable generating mechanisms—and we assume that the linguistic input to the learner (e.g., the child) is generated by a monotone computable process. But how are we to measure how well learning succeeds? Our primary measure is *prediction*—how well can the learner specify the probabilities of each possible continuation of a sentence or text? The use of prediction as a measure of learning language structure can be traced back to Shannon (1951) and has been widely used in connectionist models in language learning (e.g., Christiansen & Chater, 1994, 1999; Elman, 1990, 1993).

This prediction criterion is very difficult to meet. Intuitively, it requires that the learner must not merely uncover the phonological and syntactic rules underlying language structure, but must also master whatever other regularities determine which sentences in the corpus are generated, whether these regularities are pragmatic, semantic, or due to the influence of world knowledge. Thus, this criterion requires that the learner acquire not merely the language, but much else besides. Nonetheless, it seems intuitively plausible that if language is learned in this strong sense, it is necessarily learned in the weaker, and more natural, sense of acquiring just language-specific information, such as the grammar of the language. We shall see below that there is a precise sense in which this is true, in the section *Prediction and grammaticality*, below.

Let us frame the prediction criterion more exactly. A particular subsequence, x, is generated according to a monotone computable probability distribution μ_C. The learner is exposed to this specific x with a probability μ_C. The learner is faced with the problem of predicting how the

binary sequence will continue, i.e., what language input is to come. Let us consider this input, symbol by symbol. The basic problem is to decide the probability that sequence, x, will be followed by either a '1' or a '0'. The true probability that it will be followed by a '0' can be written, using standard notation from probability, as

$$\mu_C(0|x) = \frac{\mu_C(x0)}{\mu_C(x)}. \quad (2)$$

But, of course, the learner does not know the 'true' distribution μ_C (we will typically drop the subscript below)—because the learner does not know the language at the outset. Instead, the learner must use some other probability distribution, and hope that the predictions that it makes will approximate, to some degree, the predictions that arise from the true distribution.

2.4. The learning method: predicting by simplicity

Rather than attempting to provide a model of human language acquisition, we shall instead adopt an idealized formal model of learning. This formal model will allow an analysis of what can be learned from the linguistic input—and hence to address the issue of the poverty of the linguistic stimulus.

Specifically, the formal model is that learning follows a Simplicity Principle. The learner prefers hypotheses, theories, or patterns to the extent that they provide *simple* explanation of the data. Thus, we assume that the learner chooses the underlying theory of the probabilistic structure of the language that provides the simplest explanation of the history of linguistic input to which the learner has been exposed.

The learner can then make *predictions* about subsequent input by applying the prediction of this best (simplest) theory of the language. More accurately, as we shall see below, it makes predictions by considering the predictions of many different theories, and being influenced by each prediction to the extent that the theory that generates it provides a simple encoding of the data.

So prediction by simplicity requires finding the theory which provides the simplest explanation of the language input that has been encountered (or, more exactly, a weighted combination of explanations with the simplest explanations weighted more heavily). What does this mean in practice? A first suggestion is that the *simplest* hypothesis should be preferred. To see how this might work, consider a trivial input which consists of, for example, an initial subsequence of 1,000,000 alternations of 1 and 0, that is: 1010…1010. Intuitively, the simplest hypothesis is that the sequence continues to alternate indefinitely—leading to the prediction that the next symbol will be 1. This hypothesis is therefore favored over, for example, the intuitively more complex hypothesis that the sequence consists of 1,000,000 alternations of 1 and 0, followed by infinitely many 0 s, which would make the opposite prediction. But, taken alone, the injunction to accept the simplest hypothesis has an absurd consequence. An even simpler hypothesis, e.g., that the sequence consists of an infinite sequence of 0 s, leading to the prediction that the next symbol will therefore be a 0, will always be preferred. Such possibilities are, of course, ruled out by the constraint that the hypothesis has to be *consistent with the available data*—i.e., some hypotheses are just *too* simple. But this point itself raises difficult questions: What does it mean for a hypothesis to be consistent with the available data? Can consistency with the input be traded against simplicity of hypothesis? If so, how are simplicity and consistency with the data to be jointly optimized? The theoretical account of simplicity presented below answers these questions.

There is, however, also a more subtle difficulty: What rules out the simplest possible "vacuous" hypothesis which allows any sequence whatever—such a "hypothesis" could be interpreted as saying that "anything goes?" This hypothesis seems extremely simple; and it is also consistent with the available data. Indeed it would be consistent with *any* data, because it rules nothing out. Mere consistency or compatibility with the data is plainly not enough; the hypothesis must also, in some sense, *capture regularities* in the data. That is, it must have *explanatory power* (Harman, 1965). So we appear to be faced with the unattractive conclusion that we must somehow jointly optimize two factors, simplicity and explanatory power; and the relative influence of these two factors is unspecified.

Fortunately, there is an alternative way to proceed. This is to view a hypothesis as a way of *encoding* the data; and to propose that the hypothesis chosen is that which allows the *shortest* encoding of the data. This proposal disfavors vacuous or nearly vacuous hypotheses, that bear little or no relation to the data. These hypotheses do not help encode the data simply because they capture no regularities in the data. Focussing on using the hypothesis as a way of encoding the data also suggests an operational definition of the "explanatory power" of a hypothesis—as the degree to which that hypothesis helps provide a simple encoding of the data. If a hypothesis captures the regularities in the data (i.e., if it "explains" those regularities), then it will provide the basis for a short description of the data. Conversely, if a hypothesis fails to capture regularities in the data, then it does not provide a short description. Explanatory power is therefore not an additional constraint that must be traded off against simplicity; maximizing explanatory power is the same as maximizing the simplicity of the encoding of the data.

Measuring simplicity as brevity of encoding appears to face two problems. First, it seems that a new description language, in terms of which the linguistic or other data are to be encoded, may be required for each new hypothesis (e.g., each new grammatical theory). Second, it seems that brevity of encoding of hypotheses and data will depend on the description language chosen—and hence the predictions derived from the Simplicity Principle will likewise depend on the choice of description language.

These problems are addressed by the mathematical theory of Kolmogorov complexity (Li & Vitányi, 1997). The first problem, of needing a new language for each new type of data, is avoided by choosing a general coding language. Specifically, the language chosen is a *universal programming language*. A universal programming language is a *general purpose* language for programming a computer. The familiar programming languages such as Prolog, Java and Pascal are all universal programming languages. All these programming languages are what is called "prefix-free," that is, no syntactically correct program in the language is a proper prefix of any other syntactically correct program in the language. Moreover, the machine executing the program can determine where the program ends without having to read past the last symbol of the program. Such programs are prefix-free, effectively so, and are called "self-delimiting." For example, a language consisting of the programs "01, 001, 0001" is prefix-free, and the language "10, 100, 1000" is not prefix-free. For technical reasons we require that all universal programming languages considered in this paper are prefix-free. This is a crucial requirement for the development of the later mathematical arguments.

How can an object, such as a corpus of linguistic input, be encoded in a universal programming language such as Java? The idea is that a program in Java encodes an object if the object is generated as the output or final result of running the program. By the definition of a universal programming language, if an object has a description from which it can be reconstructed in *some* language, then it will have a description from which it can be reconstructed in the universal programming language. It is this that makes the programming language *universal*. Notice that, above, we assumed that linguistic input can be viewed as generated by a computational process (possibly mixed with a source of random input). By using a universal programming language, the learner can be sure to be able, at least in principle, to represent every such computational process.[11]

Moreover, in solving the first problem, the second problem, that different coding languages give different code lengths, is, at least partially, addressed. A central result of Kolmogorov complexity theory, the *invariance* theorem (Li & Vitányi, 1997), states that the length of the shortest description of an object, x, is invariant (up to a constant) between different universal languages.[12] The invariance theorem thus allows us to speak of *the* code length required to specify an object, x, related to a fixed choice of universal language, where this code length is, up to an additive constant, independent of the particular universal language in which the shortest code for x is written. The additive constant depends on the universal language but not on the particular object, x. The shortest code length required to specify x is defined to be its Kolmogorov complexity, $K(x)$. So, by assuming that the coding language that the cognitive system uses is universal, we can avoid having to provide a detailed account of the codes that the learner uses.

Psychologists and linguists are frequently unsettled by the invariance theorem—because such a large part of both disciplines concerns attempting to determine the nature of mental representations. Thus, any theoretical framework which treats very large classes of mental representation as equivalent may appear to be missing something important. In the present context, agnosticism concerning the specific way in which linguistic input is coded is a decided advantage—because it is possible to prove asymptotic results concerning learnability from positive evidence, irrespective of the psychological or linguistic theoretical framework adopted. Given that there is such a diversity of such frameworks currently under discussion, this lack of commitment is reassuring. On the other hand, however, the question of the *amount* of data that is required in learning does depend on representations. Thus, choice of representation can be viewed as providing the learner with a specific inductive bias, albeit a bias which will 'wash out' given the sufficient data. Nonetheless, however, the size of these biases may, for some grammatical formalisms at least, be fairly small. For example, a few hundred or at the very most, thousands, bits of information may be enough to provide a formal description of the basic formalism of phrase structure (e.g., Gazdar, Klein, Pullum, & Sag, 1985), tree-adjoining grammar (e.g., Joshi & Schabes, 1997), or categorial grammar (e.g., Steedman, 1996). We can think of these formalisms as analogous to the programming languages discussed above—and hence we can conclude that the code length differences between these formalisms will differ by at most hundreds, or perhaps thousands, of bits. Hence, the difference in inductive bias inherent in such formalisms must be fairly small (and, of course, there are, moreover, close formal relationships between them). On the other hand, the full complexity of government and binding would appear to be very much greater (e.g., Chomsky, 1981); and the complexity of the basic machinery of the minimalist program would be appear to be intermediate between these two extremes (Chomsky, 1995).

So far we have considered how to measure the complexity of individual objects. But linguistic input consists not of a single 'objects' (e.g., a single sentence, paragraph, or

[11]Strictly, a universal language can represent only the deterministic part of the mixture between deterministic and random factors assumed above to be involved in generating the corpus. This is not a substantial limitation for the learner in encoding the input, however. At any point in learning, the learner has only encountered a finite amount of data, and this finite amount of data only contains a finite amount of randomness. A universal machine can straightforwardly represent an input that contains only a finite amount of randomness (e.g., by just storing it verbatim).

[12]Crucially, this is true if all languages use the same alphabet—here, for simplicity, we assume that any coding language is, at bottom, encoded in a binary alphabet. With larger alphabets, shortest code lengths get shorter, as each choice of symbol can carry more information. Converting code lengths depending on alphabet size is straightforward—we lose no generality by restricting ourselves to a binary alphabet here.

whatever, which can be represented as a finite binary sequence) but a sequence of objects, which may be indefinitely long. How can Kolmogorov complexity be applied to measure the complexity to potentially infinite *sequences*? The only modification that we require is that complexity is measured in terms of the shortest input to a *monotone* universal machine (as discussed above), which produces a particular sequence. Thus, given that a short input to a universal monotone machine will generate a long string of 0 s, then this string has low *monotone* Kolmogorov complexity. We define the *monotone* Kolmogorov complexity of a finite sequence x as the length in bits of the shortest string such that every input that begins with this string produces an output that begins with x (the output sequence may then continue in any way at all). The monotone Kolmogorov complexity for a sequence, x, is denoted $Km(x)$. The invariance theorem holds for $Km(x)$ just as for $K(x)$ and, indeed, for finite sequences, $Km(x)$ closely approximates $K(x)$ (see Li & Vitányi, 1997, p. 285). Note also that a program on a universal monotone machine can implement a *probability distribution* over potentially infinite binary strings. The probability associated with a binary string is simply the probability the string will be generated by that random binary input—that is, the probability that it will be generated by a monkey typing random input to the computer, to pick up our earlier picture. More formally, an initial program p of length $K(\mu)$ (note that this program must be self-delimiting, so the interpreting machine can parse it) causes the universal monotone machine to start simulating the machine T_μ that transforms the following (possibly infinite) input sequence into a (possibly infinite) output sequence in such a way that the uniform distribution over the input sequences (that is, infinite sequences of 0's and 1's generated by fair coin flips) following the initial program p is transformed in the distribution μ over the output sequences. Thus, we can speak of the monotone complexity, $K(\mu)$, of a probability distribution, μ—signifying the shortest self-delimiting program on the universal machine that implements μ.[13]

We have been focussing up to now purely on finding the single simplest input program which encodes the sequence, x. But for any sequence which can be encoded at all, there will be many—in fact, infinitely many—such programs.[14] Suppose that an input program (i.e., a binary sequence) p has length $l(p)$. The probability that it will be generated by chance is $2^{-l(p)}$. Thus, the probability, $\lambda(x)$, of a sequence x being generated on a universal monotone machine is a special case of (1) above:

$$\lambda(x) = \sum_{y:M(y\ldots)=x} 2^{-l(y)}. \tag{3}$$

We shall call $\lambda(x)$ the universal monotone distribution, and, following Solomonoff (1964, 1978), we assume that the learner uses λ to predict the next linguistic input in the sequence:

$$\lambda(0|x) = \frac{\lambda(x0)}{\lambda(x)}. \tag{4}$$

This is a special case of (2) above. It is worth observing that $\lambda(x)$ is not computable—that is, there is no computable program that, for given any subsequence x, can output the probability of x according to the distribution $\lambda(.)$. $\lambda(x)$ can, however, be approximated arbitrarily closely. We shall return to the implications of these observations in the final section of this paper.

So far, we have specified the weak condition that language is generated by a monotone computable distribution, μ. We have also specified that the learner follows a Simplicity Principle—favoring hypotheses in so far as they provide brief encodings of linguistic data—and that the learner makes predictions according to a *universal* monotone computable distribution, λ. We have, furthermore, suggested that the learner's performance can usefully be assessed in terms of its ability to predict the linguistic input successfully, while allowing that another important criterion is the learner's ability to judge the grammaticality of novel sentences. We can now consider the possible effectiveness of language learnability by simplicity, from positive instances alone.

3. The prediction theorem and ideal language learning

This section rests on a key result, which we label the *Prediction Theorem*, by Solomonoff (1978). This theorem shows that, in a specific rigorous sense, the universal monotone distribution λ is reliable for predicting any computable monotone distribution μ, with very little expected error.[15]

[13]The reader may wonder why, given that we are dealing a monotone Universal Turing machine, the relevant measure for the complexity of a probability distribution is not $Km(\mu)$ rather than $K(\mu)$. The reasons are technical, but the essence is that we shall want to be able to specify a probability distribution, and then to sample from it—and to do this, we have to know when the probability distribution has been specified. Therefore, we need to be able to specify a description of the distribution, rather than a sequence which begins with a specification of the distribution (see Li & Vitányi, 1997)—that is, the code for the distribution must be self-delimiting.

[14]Consider, for example, padding a computer program with arbitrarily large amounts of null operations, in the case of a conventional computer language.

[15]Similarly it can be shown that the predictions according to the universal distribution asymptotically approximate those according to the real distribution μ, for almost all sequences (the μ-random sequences) (Li & Vitányi, 1997). As it happens, this doesn't follow from Solomonoff's result and Solomonoff's result doesn't follow from this one. Solomonoff's result states that the expected prediction error (square difference) in the nth prediction decreases faster than $1/(n \log n)$, but it doesn't state that with μ-probability1 the ratio between the conditional real probability of the nth prediction and the universal probability of the nth prediction (given the previous $n-1$ outcomes) goes to 1. The key point concerning the present result is that it must hold for almost all sequences individually, whereas Solomonoff's prediction theorem tells us something about the average taken over all sequences. This is a similar difference as that between the "Strong Law of Large Numbers" that holds for almost all infinite sequences individually and the "Weak Law of Large Numbers" that holds on average. The problem is that it is consistent with

Given the assumption, made in the previous section, that language is generated according to such a distribution, Solomonoff's result is immediately relevant to the formal problem of language acquisition. It implies that the universal monotone distribution λ is reliable for predicting what linguistic input is to come: it almost always is guaranteed to converge. For the moment, though, let us consider the result in general terms.

According to the prediction theorem, λ is a universal approximation for monotone computable probability distributions. If the learner makes predictions by using λ, the learner will rapidly close in on the 'correct' predictions of μ. Given this informal statement, it may seem that λ may sound too good to be true—and indeed it may seem conceptually impossible that a single distribution can *simultaneously* approximate every one of the entire class of computable probability distributions, because these distributions will themselves be so diverse. To see how this is possible, let us state Solomonoff's result more precisely.

Suppose that a sequence of $n-1$ binary values are generated by a computable data-generating mechanism, associated with a probability distribution, μ, in the way described in the previous section. Let us call this sequence of $n-1$ values x. Given this sequence, we can ask how closely the prediction according to μ agrees with the predictions from the prior, λ. Specifically, we measure the difference in these predictions by the square of difference in the probabilities that μ and λ assign to the next symbol being 0.[16] Formally, this difference is

$$\text{Error}(x) = (\lambda(0|x) - \mu(0|x))^2. \tag{5}$$

$\text{Error}(x)$ measures how good an approximation $\lambda(0|x)$ is to $\mu(0|x)$—but its value clearly depends on which previous sequence of items, x, has been generated. To get a general comparison between λ and μ, we need to take into account the various sequences x that $\mu(x)$ might have generated. Moreover, we weight these sequences by the probability $\mu(x)$ that they were generated by the true distribution, μ.

$$s_n = \sum_{x:l(x)=n-1} \mu(x)\text{Error}(x), \tag{6}$$

s_n is thus the *expected* value of the squared error between the predictions of λ and μ on the nth prediction. The smaller the s_n, the better λ predicts μ.

This weighting by the actual distribution, μ, reflects the fact that we would intuitively view λ as a good approximation to the true distribution μ if it assigns similar probabilities to events which are actually likely to occur (according to μ). It does not matter whether μ and λ disagree on cases that never arise in practice (i.e., where $\mu(x)$ is 0). This weighting by the actual distribution will be important below, when we apply these ideas to language learning. Specifically, in assessing how well a learner has learned the structure of a language, there will be considerable weight attached to linguistic material which might actually be said; and little weight attached to sentences (e.g., a sentence containing 1000 clauses linked by *and*) which will never actually be produced.

Finally, the expected prediction performance over the entire sequence is just:

$$\sum_{j=1}^{\infty} s_j. \tag{7}$$

We shall use this measure as our overall measure of predictive success. To get a feel for the meaning of $\sum_{j=1}^{\infty} s_j$, consider the case where the expected value of the sum square difference between two computable probability distributions μ_1 and μ_2 is always greater than some constant d, where d may be arbitrarily small. In this case, $\sum_{j=1}^{\infty} s_j$ is at least $(\infty)(d)$ which is, of course, ∞. But, remarkably, Solomonoff's Prediction Theorem shows that, in relation to the distributions λ and μ considered here, the sum $\sum_{j=1}^{\infty} s_j$ has a limit bounded by a constant, and hence that as the amount of data increases, the expected prediction error tends to 0. That is, given enough data, expected prediction should be almost perfect—using the universal distribution λ the learner should accurately be able to learn, in an approximate sense, any true computable distribution μ. Specifically, the following result holds:

Prediction Theorem. (Solomonoff, 1978) *Let μ be a computable monotone distribution. Then,*

$$\sum_{j=1}^{\infty} s_j \leqslant \frac{\log_e 2}{2} K(\mu). \tag{8}$$

We shall consider below how the Prediction Theorem can be related to language acquisition, but first, we show how Solomonoff's remarkable theorem can be proved.[17] The proof has four steps.

The first, and crucial, step is to show that, for any finite or infinite computable data x:

$$\log_2 \frac{\mu(x)}{\lambda(x)} \leqslant K(\mu). \tag{9}$$

This puts an upper bound on how much $\mu(x)$ can exceed $\lambda(x)$. Intuitively, it implies that if x is probable according to

(*footnote continued*)
Solomonoff's result that $\mu(0|x) = 0$ infinitely often which prevents the ratio $\lambda(0|x)/\mu(0|x)$ from going to 1 in the limit. Nonetheless, Solomonoff's result has a speed-of-convergence estimate that is quite strong (but only holds for the average) while the convergence law has no speed-of-convergence estimate although it guarantees convergence with probability 1.

[16]We could, of course, equally well consider the difference in the probability that the next symbol is a 1, with no substantive change to the proof.

[17]We here follow the spirit and much of the notation of Li and Vitányi (1997) treatment, which is based on a proof suggested by Peter Gács. Solomonoff's original proof is quite different. We have also reworked the proof in order to reduce it to its essentials as far as possible, and to provide a self-contained presentation, not presupposing knowledge of algorithmic probability theory (e.g., Zvonkin & Levin, 1970) or the general theory of Kolmogorov complexity (Li & Vitányi, 1997).

μ, it is also reasonably probable according to the universal prior probability λ.

The second step is to show that (9) implies a bound on a measure of similarity between the distributions μ and λ over the set of computable data strings x. This measure of similarity is Kullback–Liebler distance $D(\mu\|\lambda)$:

$$D(\mu||\lambda) \leqslant K(\mu). \tag{10}$$

The third step breaks up this similarity measure over the distribution over whole strings x into an infinite sum of Kullback–Liebler similarity measures over each of the j positions in the string $D_j(\mu\|\lambda)$. This step is conceptually straightforward, but algebraically complex:

$$\sum_{j=1}^{\infty} D_j(\mu||\lambda) = D(\mu||\lambda). \tag{11}$$

The final step makes the connection between the Kullback–Liebler measure of similarity and the measure of similarity with which we are primarily concerned: sum-squared error.

$$\sum_{j=1}^{\infty} s_j \underset{\text{step 4}}{\leqslant} \frac{\log_e 2}{2} \sum_{j=1}^{\infty} D_j(\mu||\lambda). \tag{12}$$

We can now put the last three steps together to obtain the final result:

$$\sum_{j=1}^{\infty} s_j \underset{\text{step 4}}{\leqslant} \frac{\log_e 2}{2} \sum_{j=1}^{\infty} D_j(\mu||\lambda) = \underset{\text{step 3}}{\frac{\log_e 2}{2}} D(\mu||\lambda) \underset{\text{step 2}}{\leqslant} \frac{\log_e 2}{2} K(\mu) \tag{13}$$

thus proving the theorem. We now prove each step in turn.

Step 1: To prove

$$\log_2 \frac{\mu(x)}{\lambda(x)} \leqslant K(\mu). \tag{9'}$$

Consider a universal monotone machine, U. Because U is universal, for any monotone machine W, there will be a finite string, p, which 'programs' U to behave like W.[18] That is, for any monotone machine W, there will be a program p (in fact, many such programs), such that for all x, $U(px) = W(x)$. Since U must parse px into p and x it must be able to detect the end of p; that is, p must be self-delimiting. As noted above, the probability of randomly generating a binary program, p, of length $l(p)$, is $2^{-l(p)}$. This means that short programs have the highest probability.

Let us therefore focus on the shortest and hence most probable program for W in U; if there are several programs which tie for the shortest length, we choose one of them arbitrarily. The length of this shortest self-delimiting program is $K(W)$ by the definition of prefix-complexity, K. Each monotone machine is associated with a distribution over all its possible output sequences, as defined above. If W is associated with the true probability distribution μ, then we can write $K(\mu)$ to denote the length of the shortest program which generates μ.

Now consider a string x, with probability $\mu(x)$. What can we say about $\lambda(x)$? By the considerations above, we know that one input (of many inputs) to λ which will generate x is as follows: first a program of length $K(\mu)$ which converts U into W (and hence λ into μ) followed by any of the inputs to W which produce x. The probability of the first part is the probability of a specific binary sequence of length $K(\mu)$, which is $2^{-K(\mu)}$. The probability that the second part of the sequence then generates x is, by definition, $\mu(x)$. Thus the probability of both parts of the sequence is the product of these two:

$2^{-K(\mu)}\mu(x)$. This is just one way of obtaining x in U (there are infinitely many other programs which produce any given output, of course), which means that the probability of this single program cannot be greater than $\lambda(x)$, the overall probability of obtaining output x from U. Thus,

$$2^{-K(\mu)}\mu(x) \leqslant \lambda(x). \tag{14}$$

Rearranging, we get

$$\frac{\mu(x)}{\lambda(x)} \leqslant 2^{K(\mu)}. \tag{15}$$

Taking logs in base 2 of both sides of the inequality proves Step 1.

Step 2: To prove

$$D(\mu||\lambda) \leqslant K(\mu). \tag{10'}$$

Let us introduce a measure of the similarity between two probability distributions, which can be related to, but is easier to deal with, than sum-squared difference, defined above. This measure is Kullback–Liebler divergence, $D(P\|Q)$. This measure originates in information theory. It measures the expected amount of information that is wasted in transmitting a message x which is actually generated by a probability distribution P, but is encoded using a code which is instead optimally adjusted to transmit messages generated by probability distribution Q. The waste arises because an optimal code assigns short codes to probable items, and longer codes to less probable items. Thus, if P and Q are very different, then codes will be assigned in an inappropriate way. Specifically, short codes will be used for items which are probable according to Q, but which may not be probable according to the actual distribution P, and vice versa. When $P = Q$, there is, of course, no waste at all, and $D(P\|P)$ is therefore 0. Moreover, if $P \neq Q$ the expected waste is positive, so that $D(P\|Q) \geqslant 0$—and the amount of waste measures how similar or different the two probability distributions are.[19]

[18] Strictly, we stipulate that this program is *self-delimiting*, which means that it is clear when the end of the program has been reached, and hence when the data input to the program begins. This apparently minor point actually has substantial mathematical consequences, which bedeviled early attempts to formalize these ideas (e.g., Solomonoff, 1964).

[19] Some theories of similarity in cognitive science presuppose that similarity must be symmetrical. That is, A must be exactly as similar to B as B is to A. But Kullback Liebler divergence is not symmetrical. Hence, from the perspective of these accounts, Kullback–Liebler distance can be

The Kullback–Liebler divergence between probability distributions P and Q is defined:[20]

$$D(P||Q) = \sum_x P(x) \log_2 \frac{P(x)}{Q(x)}. \tag{16}$$

Let us now consider the Kullback–Liebler divergence between the distributions $\mu(x)$ and $\lambda(x)$, where x ranges over (possibly infinite) output sequences:

$$D(\mu||\lambda) = \sum_x \mu(x) \log_2 \frac{\mu(x)}{\lambda(x)}. \tag{17}$$

Applying (9), we have

$$D(\mu||\lambda) \leqslant \sum_x \mu(x) K(\mu) \leqslant K(\mu), \tag{18}$$

where the second inequality follows because $\mu(x)$ is a semi-measure, i.e., $\sum_x \mu(x) \leqslant 1$. This proves Step 2.

Step 3: To prove

$$\sum_{j=1}^{\infty} D_j(\mu||\lambda) = D(\mu||\lambda). \tag{11′}$$

We have seen how Kullback–Liebler divergence can be defined over distributions of entire (possibly infinite) sequences. It will turn out to be useful to relate this to the Kullback–Liebler divergence at each location in the sequence.

A useful intuition concerning how this works is as follows. $D(\mu\|\lambda)$ measures the expected amount of 'wasted' information required to send a randomly selected sequence generated by the distribution μ, using codes which are optimal relative to the assumption that the distribution is λ, over and above the expected amount of information required if the codes were optimized to the true distribution, μ. Suppose we consider the expected amount of information wasted in transmitting the first symbol; then the expected amount of information wasted in transmitting the second symbol; and so on. These quantities correspond to Kullback–Liebler divergences, defined over each symbol in turn. It seems plausible that the sum of the expected amounts of information wasted in transmitting each symbol should be equal to the total amount of information wasted in transmitting the entire sequence—this is the intuitive content of the result that we are aiming to prove.

To put this more exactly, we need to express the expected amount of information wasted at symbol j. Suppose that the sequence of symbols from 1 to $j-1$ is x. According to μ, the probabilities of the next symbols are given by $\mu(\cdot|x)$. Similarly, according to λ, the probabilities of the next symbols are given by $\lambda(\cdot|x)$.

Then, using standard Kullback–Liebler distance regarding the outcomes for the jth symbol, we have

$$D_j(\mu(\cdot|x)||\lambda(\cdot|x)) = \sum_{a=0,1} \mu(a|x) \log_2 \frac{\mu(a|x)}{\lambda(a|x)}. \tag{19}$$

The expected value of this term with respect to the true distribution $\mu(.)$ requires weighting it by $\mu(x)$, the probability that the first $j-1$ symbols in the sequence are x according to the true distribution μ. Thus, the expected amount of wasted information in encoding the jth symbol using λ instead of μ, which we shall denote by $D_j(\mu\|\lambda)$ is

$$D_j(\mu||\lambda) = \sum_{x:l(x)=j-1} \mu(x) D_j(\mu(\cdot|x)||\lambda(\cdot|x)). \tag{20}$$

Here we have merely defined the terms in the conjecture above, and explained the intuition behind it. That is, the amount of information wasted in transmitting a sequence by using a code optimized to the 'wrong' probability distribution is the same, whether the sequence is encoded all at once, or symbol by symbol. A rigorous derivation that substantiates this intuition is given in Appendix B.

Step 4: To prove

$$\sum_{j=1}^{\infty} s_j \leqslant \frac{\log_e 2}{2} \sum_{j=1}^{\infty} D_j(\mu||\lambda). \tag{12′}$$

We have shown that the learner's distribution λ is similar to any computable distribution μ, where similarity is measured by Kullback–Liebler distance. Moreover, we have shown how the expected divergence between the distributions over infinite sequences can be converted to a sum of the expected divergences at each location in the series. It remains to relate Kullback–Liebler distance to the familiar measure of goodness of prediction with which we began: the expected sum-squared error between μ and λ.

The key to doing this is the following result, which applies to arbitrary distributions P and Q that can take just two values 0 and 1 (the proof is given in Appendix C).

$$(P(0) - Q(0))^2 \leqslant \frac{\log_e 2}{2} D(P||Q). \tag{21}$$

It immediately follows that the same result holds if P and Q are conditional on a previous string, x:

$$(P(0|x) - Q(0|x))^2 \leqslant \frac{\log_e 2}{2} D(P(\cdot|x)||Q(\cdot|x)). \tag{22}$$

Substituting μ and λ and for P and Q, and using the definition of Error(.) in Eq. (5), we obtain

$$\begin{aligned} \text{Error}(x) &= (\mu(0|x) - \lambda(0|x))^2 \\ &\leqslant \frac{\log_e 2}{2} D(\mu(\cdot|x)||\lambda(\cdot|x)). \end{aligned} \tag{23}$$

(*footnote continued*)
related to similarity only at a metaphorical level. We nonetheless use the term 'similarity' in relation to Kullback–Liebler distance here, for clarity, without intending any particular stand on these issues (see, e.g., Chater & Vitányi, 2003; Hahn & Chater, 1998).

[20]Kullback–Liebler divergence is sometimes defined using logs in base e, rather than base 2. This leads to some minor differences between statements of results here and those in Li and Vitányi (1997).

Using the definition of s_j (Eq. (6)),

$$s_j = \sum_{x:l(x)=j-1} \mu(x)\text{Error}(x)$$
$$\leqslant \sum_{x:l(x)=j-1} \mu(x)\frac{\log_e 2}{2} D(\mu(\cdot|x)||\lambda(\cdot|x))$$
$$\leqslant \frac{\log_2 2}{2} D_j(\mu||\lambda). \quad (24)$$

We now take the expected sum squared error over all symbols in the sequence, which immediately gives Eq. (12), and hence proves Step 4.

Having proved Steps 1–4, we have hence completed the proof of the Prediction Theorem.

The Prediction Theorem provides a counterweight (alongside more specific positive learnability results, e.g., Feldman, 1972; Horning, 1969; Pitt, 1989; van der Mude & Walker, 1978) to some interpretations of Gold's (1967) negative results concerning the apparently limited conditions under which languages can be learned in the limit from positive evidence. It shows that learning by simplicity can, in principle, be expected to converge to the correct conditional probabilities in predicting subsequent linguistic material. Intuitively, if an ideal learner can predict accurately, it seems that it must be able to learn a great deal about the range of linguistic, pragmatic, social and environmental factors which influence the linguistic input that is received. This appears to imply that the learner must know a good deal about the specifically *linguistic* structure of the language.

It is appropriate to ask whether this intuition can be backed up with a quantitative measure of how well the learner must acquire specifically linguistic information. The results in the next section shows that this can be done, by putting an upper bound on the number of 'grammaticality' errors that the learner can make in the course of predicting the linguistic input.

4. The ideal learning of grammaticality judgments

A straightforward test of the learner's ability to distinguish grammatical from non-grammatical linguistic input: Suppose that the learner has to 'guess' the next word in the text at each point. How often does the ideal learner overgeneralize and guess continuations that are ungrammatical? And how often does it undergeneralize, and erroneously rule out continuations which are, in reality, acceptable in the language? We consider bounds on each type of error in turn.

4.1. Overgeneralization errors

In overgeneralization, linguistic sequences are allowed by the learner's probability distribution (i.e., they are viewed as grammatical by the learner), but they are not allowed by the true grammar. We wish both to measure, and to attempt to put limits on, the amount of overgeneralization that learning according to the Simplicity Principle will involve (here, we ignore the possibility of "performance error" by speakers producing the linguistic input to the learner—we assume that the linguistic input consists purely of grammatical sentences).

In the discussion of both overgeneralization and undergeneralization, it is convenient to consider language input as a sequence of words,[21] rather than coded as a binary sequence. Of course, a binary sequence is, by stipulation, simply a particular way of encoding words—words are encountered one-by-one, and each word stands in one-to-one correspondence with a binary string. Thus, each possible corpus of language, viewed as a sequence of words, stands in a one-to-one correspondence with a possible binary string; and the probabilities of each corpus of words are identical to the probabilities associated with the corresponding binary strings. Thus, instead of dealing with distributions over finite and infinite *binary sequences*, μ, and the learner's universal approximation, λ, we shall deal with corresponding distributions defined over finite and infinite sequences of *words*. We shall call these corresponding distributions P_μ and P_λ.

Suppose that the learner has seen a specific corpus, x, of $j-1$ words. Suppose that the learner has a probability $\Delta_j(x)$ of erroneously guessing that the next (i.e., the jth) word in the input is a word which is actually not allowed by the grammar. In symbols, we can define:

$$\Delta_j(x) = \sum_{\substack{k:xk \text{ is ungrammatical,} \\ l(x)=j-1}} P_\lambda(k|x). \quad (25)$$

That is, $\Delta_j(x)$ is the amount of probability that the learner devotes to grammatically impossible overgeneralization on the jth word. Because we assume that the linguistic input contains no noise, ungrammatical continuations have zero probability of occurring.

The probability $\Delta_j(x)$ will, of course, depend on the specific x that has been encountered. The *expected* value of $\Delta_j(x)$, which we shall write $\langle \Delta_j \rangle$, is defined as follows:

$$\langle \Delta_j \rangle = \sum_{x:l(x)=j-1} P_\mu(x)\Delta_j(x). \quad (26)$$

Our goal is to put some bound on the expected number of overgeneralization errors throughout the corpus, i.e., to put a bound on $\sum_{j=1}^{\infty} \langle \Delta_j \rangle$. The following *Overgeneralization Theorem* holds.

[21] Nothing theoretically substantial rests on the choice of the word as the unit of choice. The important point here is that language is considered as a sequence of a finite number of linguistically significant and separate chunks. The arguments below would equally well go through if we assumed that language input were coded in terms of phonemes, morphemes or syllables.

Overgeneralization theorem. *Where* $\langle \Delta_j \rangle$ *is defined as above*,

$$\sum_{j=1}^{\infty} \langle \Delta_j \rangle \leqslant \frac{K(\mu)}{\log_e 2}. \tag{27}$$

That is, the expected amount of probability devoted by the learner to overgeneralizations, in the course of encountering an infinite corpus, sums to a finite quantity. Thus, the typical degree of overgeneralization, as the corpus increases in size, must go to 0.

Proof. The proof has two parts. The first part concerns how the waste of *probability*, $\Delta_j(x)$, due to overgeneralization after seeing a sequence x, inevitably leads to a waste of *information*. This information is quantified by the Kullback–Liebler divergence between P_μ and P_λ, which can later on be related to $K(\mu)$. But this leaves a crucial gap—it deals with $\Delta_j(x)$ for some particular x; but it says nothing about $\langle \Delta_j \rangle$, the *expected* amount of probability wasted by the learner, averaged across all x. The second part of the proof fills in this gap, and hence provides the required bound on $\sum_{j=1}^{\infty} \langle \Delta_j \rangle$. To finish the proof, we also need to relate the results from these two steps to some of the analysis we have described above, in proving the Prediction Theorem.

The first part of the proof begins by considering the following scenario. Suppose that the learner uses its probability distribution P_λ to *encode* the output from the true underlying distribution, P_μ. After the sequence, x, of $j-1$ words has been encountered, we can ask: What is the expected amount of wasted information in encoding the jth item? Such waste is inevitable, because the learner is using codes which are optimized to the learner's distribution (i.e., $P_\lambda(.|x)$), rather than to the true (but from the learner's point of view, unknown) distribution (i.e., $P_\mu(.|x)$). The key underlying intuition is that, to the extent that the learner has a tendency to overgeneralize, the learner must necessarily waste a certain amount of information. This is because the learner encodes items as if some continuations are possible, where in reality they are not possible. This means that some code length must be 'used up' in specifying the actual continuation in order to rule out these continuations. The greater the degree to which the learner overgeneralizes, the greater the amount of wasted information.

Suppose, then, that the learner has a particular $\Delta_j(x)$. How much wasted information must follow from this wasted probability? The minimum level of wasted information is achieved as follows.[22] Assume that, for all the other lexical items, k, which are possible continuations, the probability assigned by the learner to this continuation, $P_\lambda(k|x))$, is just $(1-\Delta_j(x))$ times the true probability $P_\mu(k|x)$. Thus, a certain amount of 'probability' is wasted by the learner, on continuations that are impossible; but otherwise the probabilities of all the possible continuations are correct, except that they have to be appropriately re-scaled. What is the expected amount of waste that occurs by encoding the actual continuation in terms of the learner's $P_\lambda(k|x))$, rather than the true $P_\mu(k|x)$, using this maximally efficient 're-scaled' encoding? Applying Kullback–Liebler divergence:

$$\begin{aligned} D(P_\mu(\cdot|x)||P_\lambda(\cdot|x)) &\geqslant \sum_k P_\mu(k|x) \log_2 \frac{P_\mu(k|x)}{P_\lambda(k|x)} \\ &= \sum_{xk \text{ ungrammatical: } P_\mu(k|x)=0} P_\mu(k|x) \log_2 \frac{P_\mu(k|x)}{P_\lambda(k|x)} \\ &\quad + \sum_{xk \text{ grammatical; } P_\mu(k|x)\geqslant 0} P_\mu(k|x) \log_2 \frac{P_\mu(k|x)}{(1-\Delta_j(x))P_\mu(k|x)}. \end{aligned} \tag{28}$$

The first term is 0, because $P_\mu(k|x)$ is zero for ungrammatical continuations. Simplifying the second term, we obtain[23]

$$\log_2\left(\frac{1}{(1-\Delta_j(x))}\right) \sum_{xk \text{ grammatical, } P_\mu(k|x)\geqslant 0} P_\mu(k|x). \tag{29}$$

Because the input continues in some way or other, $\sum_{xk\text{grammatical, } P_\mu(k|x)\geqslant 0} P_\mu(k|x) = 1$, and hence we can conclude that

$$D(P_\mu(\cdot|x)||P_\lambda(\cdot|x)) \geqslant \log_2\left(\frac{1}{(1-\Delta_j(x))}\right). \tag{30}$$

This is the minimum expected amount of waste that accrues for a particular guess, with probability $\Delta_j(x)$ of the learner guessing an ungrammatical continuation.

We have considered a particular x. We now average over all the possible sequences of $j-1$ words, to get the expected amount of information loss on encoding the jth item, which is denoted by $D_j(P_\mu(\cdot|x)||P_\lambda(\cdot|x))$ (using the definition in Eq. (20) above). Thus, we obtain

$$\begin{aligned} D_j(P_\mu(\cdot|x)||P_\lambda(\cdot|x)) &\geqslant \sum_{x:l(x)=j-1} P_\mu(x) \log_2\left(\frac{1}{1-\Delta_j(x)}\right) \\ &= \left\langle \log_2\left(\frac{1}{1-\Delta_j(x)}\right)\right\rangle. \end{aligned} \tag{31}$$

This completes the first part of the proof.

The second part of the proof shows how the above result can be applied to put a bound on $\sum_{j=1}^{\infty} \langle \Delta_j \rangle$. Log is a concave function, and we can therefore use the standard result that for expectations over an arbitrary random variable, z (where $z>0$):

$$\log_2 \langle z \rangle \geqslant \langle \log_2 z \rangle. \tag{32}$$

This implies that

$$\begin{aligned} -\langle \log_2 z \rangle &\geqslant -\log_2 \langle z \rangle, \\ \left\langle \log_2 \tfrac{1}{z} \right\rangle &\geqslant \log_2 \tfrac{1}{\langle z \rangle}. \end{aligned} \tag{33}$$

[22] See Appendix D for a proof of this 're-scaling lemma.'

[23] Note that this formula allows for the possibility that there are grammatical sentences which have zero probability of being heard.

if we then substitute in $1-\Delta_j(x)$ for z, we obtain

$$\left\langle \log_2 \frac{1}{1-\Delta_j(x)} \right\rangle \geqslant \log_2 \frac{1}{\langle 1-\Delta_j(x)\rangle} = \log_2 \frac{1}{1-\langle \Delta_j(x)\rangle}. \tag{34}$$

This is a crucial part of the second step in the proof—we have now introduced the *expected* value, $\langle \Delta_j(x)\rangle$, across all possible x. We can now get at $\langle \Delta_j(x)\rangle$ more directly, but replacing the log expression on the right hand side of the inequality using a Taylor expansion:

$$\begin{aligned} &\log_2\left(\frac{1}{(1-\langle \Delta_j(x)\rangle)}\right) \\ &= \log_2 e\left(\langle \Delta_j(x)\rangle + \frac{\langle \Delta_j(x)\rangle^2}{2} + \cdots + \frac{\langle \Delta_j(x)\rangle^m}{m} + \cdots\right) \\ &\geqslant \langle \Delta_j(x)\rangle \log_2 e. \end{aligned} \tag{35}$$

Stringing together the inequalities (31), (34) and (35), we obtain

$$D_j(P_\mu(\cdot|x)||P_\lambda(\cdot|x)) \geqslant \langle \Delta_j(x)\rangle \log_2 e. \tag{36}$$

So far we have only considered the probability of overgeneralization, and consequent waste of information, for the jth word in the corpus. We now sum over all j. The left hand side of Eq. (36) immediately simplifies, using the result above (Eq. (20)) that $\sum_{j=1}^{\infty} D_j(\mu||\lambda) = D(\mu||\lambda)$. This gives

$$D(P_\mu||P_\lambda) \geqslant \langle \Delta_j(x)\rangle \log_2 e. \tag{37}$$

In this section, we have so far worked with probability distributions P_μ and P_λ over sequences of words, rather than with the familiar μ and λ, which are defined over binary sequences. We can now relate the present discussion back to the binary analysis. By stipulation, there is a one to one correspondence between possible binary states and sequences of words.[24] There is therefore also a direct correspondence between the probabilities of these corresponding states. The probability of generating a particular word sequence is the same as the probability of generating the corresponding binary sequence. Information-theoretic measures, such as Kullback–Liebler distance, are of interest precisely because they are independent of the details of the coding scheme used to represent a probability distribution. Thus, it makes no difference whether the probability distributions are defined over strings of words (like P_μ and P_λ) or are defined over the corresponding binary strings (like μ and λ). Hence,

$$D(P_\mu||P_\lambda) = D(\mu||\lambda) \leqslant K(\mu), \tag{38}$$

where the right hand inequality follows from (20) above. Putting (37) and (38) together gives the result

$$\sum_{j=1}^{\infty} \langle \Delta_j\rangle \leqslant \frac{K(\mu)}{\log_e 2}. \qquad \square \tag{39}$$

The intuitive significance of the overgeneralization theorem can be thought of in the following way. Suppose that the language learner were to continually attempt to guess the next word of every linguistic interchange. If the learner follows the Simplicity Principle, and makes predictions according to the distribution P_λ (or, equivalently, according to λ over a binary code), then the expected number of times that the learner will make a prediction that violates the grammar of the language has a finite bound, even on an infinite corpus. This implies, for example, that, for a linguistic input of n words, the expected average number of overgeneralization errors can be no more than: $K(\mu)/n \log_e 2$. Thus, if we consider a sufficiently large corpus (i.e., we increase n), the average expected number of overgeneralization errors tends to zero.

4.2. Undergeneralization errors

In undergeneralization, a sentence, s, is allowed by the true grammar, but it disallowed by the learner's probability distribution. If this were to occur, after hearing a prior sequence of words x, a word, k, would be encountered which the learner had assigned a probability of 0. The learner would have undergeneralized, by assuming that the language is more restrictive than it in fact is.[25]

For a learner using the Simplicity Principle, however, such undergeneralizations *never* occur. This apparently remarkable result can be understood intuitively as following simply from the fact that the learner's probability distribution, λ, corresponds to a *universal* monotone computer. Any computable output (including any corpus of language generated by a monotone computable process) therefore has a non-zero probability of being generated by this universal machine—because a universal machine, by definition, can simulate the computable process that

[24] Strictly, this is true for binary states with non-zero probability of occurrence. We assume that all and only the binary strings that can be generated are sequences of words—the whole point of the binary code is to encode language input.

[25] Note that the learner might undergeneralize not only because of an underestimation of which sentences are grammatical. The learner might, instead, assume that a certain sentence is impossible for a variety of other reasons. For example, the learner might wrongly assume that people can only produce center-embedded sentences of depth one—this could be viewed as an incorrect estimation of people's short-term memory constraints, rather than a misconstrual of the grammar. In more general terms, to the extent that a distinction between linguistic competence and linguistic performance can be made (Chomsky, 1965), the learner may undergeneralize with respect to either competence or performance. The bounds that we develop here apply to undergeneralization of both kinds; and hence automatically provide bounds on undergeneralizations of linguistic competence, which are of most interest to linguists. Hence, we need not consider the difficult questions concerning how, if at all, the competence/performance distinction can be made precise (though see Christiansen & Chater, 1999).

generated this output. There is therefore a non-zero probability that a program that simulates this computational process will be generated by chance.[26]

But further reflection suggests that the problem of undergeneralization has not really been ruled out effectively by the analysis above. So far we have ruled out the possibility that the learner assumes a continuation to be impossible, when it is actually possible; but it seems relevant also to consider the case where the learner drastically underestimates (perhaps by a vast factor) the probability that a sentence might occur. In this case, the true distribution might allow that a continuation (e.g., *dogs* after the context *raining cats and*…) is actually rather common; whereas the learner believes that it is so infinitesimally probable that it is unlikely to occur in the entire history of the universe. Such a learner would seem, intuitively, to be making an undergeneralization error (and a rather blatant one!); but such errors will not be detected by the previous criterion, as the learner believes the probability of the continuation to be non-zero.

To address this concern, let us therefore consider a 'soft' version of undergeneralization. Suppose, as before, that the sequence of words encountered by the learner is generated according to a computable probability distribution P_μ, and that the learner attempts to predict this sequence by a universal probability distribution P_λ. As usual, we denote the sequence of the initial $j-1$ words that the learner encounters by x, and let us call the jth word, k. If the learner undergeneralizes on word k by a factor f, this means that the learner underestimates the probability that k will occur after x by a factor f. That is, $P_\lambda(k|x)f \leqslant P_\mu(k|x)$. What is the probability that the k that is chosen according to the true distribution is a word on which the learner undergeneralizes, given the preceding sequence, x? This probability, which we denote $\Lambda_j(x)$, can be expressed as

$$\Lambda_j(x) = \sum_{k: f.P_\lambda(k|x) \leqslant P_\mu(k|x)} P_\mu(k|x). \tag{40}$$

The *expected* probability, $\langle \Lambda_j \rangle$, with which this occurs on the jth item is expressed:

$$\langle \Lambda_j \rangle = \sum_{x: l(x)=j-1} P_\mu(x)\Lambda_j(x). \tag{41}$$

Our goal is to put some bound on the expected number of undergeneralization errors throughout the corpus, i.e., $\sum_{j=1}^{\infty} \langle \Lambda_j \rangle$. The following result can be derived (see Appendix F for a proof):

Soft undergeneralization theorem.

$$\sum_{j=1}^{\infty} \langle \Lambda_j \rangle \leqslant K(\mu) \frac{1}{\log_2 f/e} \tag{42}$$

(*so long as* $f > e$).

The theorem implies that the expected number of 'soft' undergeneralizations is bounded by a constant, even for an infinitely long sequence of linguistic input. As with overgeneralizations, the upper bound is proportional to the complexity of the underlying probabilistic mechanism generating the language (including, presumably, the grammar of the language). Moreover, the more severe the criterion for an undergeneralization (the greater the value of f), the fewer such undergeneralizations can occur.

We have shown that, if language is generated by an arbitrary computable probability distribution, P_μ, and the learner employs the universal distribution P_λ, the expected number of over- and under-generalizations that the learner makes will be bounded by a constant, over an infinitely long linguistic input.

Thus, in testing grammaticality judgments by prediction, as discussed above (and assuming the highly idealized case where linguistic input consists *only* of grammatical sentences), the learner can, in the limit, make highly accurate grammaticality judgments.

5. The ideal learning of language production

So far we have presented two results. First, we have shown that learning using a Simplicity Principle can be used to successfully predict linguistic input, in the asymptote; this result arises directly from Solomonoff's (1978) Prediction Theorem. Second, we showed that the Prediction Theorem has implications for the ability to learn to make grammaticality judgments from positive evidence alone. Roughly, the logic of the argument was to show how a learner that can predict effectively can use this ability to make grammaticality judgments; and hence to use the result concerning the quality of prediction to provide an insight to the quality of grammaticality judgments.

It might appear, however, that a more challenging task for the learner is not merely to *judge* whether sentences that it hears are grammatical, but to successfully *produce* sentences of its own. Fortunately, it is possible to show that by learning using the universal distribution, λ, the learner can also produce language effectively, in the asymptote.

To see how this works, we can imagine that our ideal learner has been exposed to a large corpus of linguistic input, involving conversation between other speakers. The learner's goal is to be able to join the conversation with linguistic outputs of its own—in a way that is indistinguishable from the linguistic outputs of other speakers. If the learner is able to blend in successfully with such conversation, then it must have learned to produce language in conformity with the grammatical, semantic, and other, regularities respected by other speakers. Of course, the mere ability to blend in with other speakers is a limited goal—in practice, language learners wish to be able produce language that does much more: that reflects their own specific beliefs and utilities. We consider an aspect of how this ability can be learned, by learning to map represenatations of linguistic meaning and linguistic form, in future work.

[26] A proof is given in Appendix E.

Let us consider some particular contribution, y, that our ideal learner decides upon, after hearing a linguistic corpus, x (for convenience, assume these are encoded as binary strings). The probability that the sentence has this continuation, if the sequence continues the corpus generated by the existing speakers, is $\mu(y|x)$. The learner generates utterances instead by the same distribution that it uses in prediction, i.e., with probability $\lambda(y|x)$. The learner blends in, to the extent that $\lambda(y|x)$ is a good approximation to $\mu(y|x)$.

The following result ensures that the match is a good one (Li & Vitányi, 1997, Theorem 5.2.2). Where μ is a probability distribution (strictly, a semi-measure) generated by a monotone computable process, and λ is the universal distribution (used by the learner), then for any finite sequence y, then as the length of sequence x tends to infinity:[27]

$$\frac{\lambda(y|x)}{\mu(y|x)} \to 1 \tag{43}$$

with a probability converging to 1 for fixed y and as the length of x increases (provided there is an $\varepsilon > 0$ such that $\mu(y|x) > \varepsilon$ for all y and x involved). Interpreting (43) in the context of language production, this means that, in the asymptote, the learner will blend in arbitrarily well. The probability of the learner producing any continuation of the conversation will tend towards the probability of that continuation being made by another speaker. In particular, this means that there will not be sentences that the other speakers might say with some significant probability, but which the learner is incapable of saying; and conversely that everything that the learner might say with significant probability will be something that the other speakers might have said. Thus, in the asymptote, the learner can speak the language indistinguishably from the speakers in the language community in which the language was learned.

6. The poverty of the stimulus reconsidered

We have shown that, under quite broad assumptions about the linguistic input over which learning occurs, there is enough information in positive input alone to learn a good deal about a language. In this section, we briefly consider the application of these results to a concrete linguistic discussions; we reconsider the relationship of the present results to the logical and construction-specific versions of the poverty of the stimulus argument, as discussed earlier, and we also outline open questions for future research.

[27]Strictly, this theorem does not hold for all sequences xy; but the probability that the theorem holds tends to 1, as the length of x tends to infinity. Thus, the 'pathological' sequences where the theorem does not hold will do not arise too often in practice.

6.1. Implications for theories of language acquisition

To make the implications of this theorem linguistically concrete, note that our results have direct implications for the learnability, from positive evidence, of any specific principle of grammar. Suppose, for example, we consider the subtle principles of government and binding (e.g., Chomsky, 1981, 1986) that are presumed to explain that 44a and 44b are possible in English, but that 44c and 44d are not:

(a) John is too stubborn [to talk to].
(b) John is too stubborn [to expect [anyone to talk to]].
(c) *John is too stubborn [to visit [anyone who talked to]].
(d) *John is too stubborn [to ask anyone [who talked to]].

(44)

The principles underlying these and many related phenomena (Chomsky, 1986) seem to be enormously intricate. It might therefore be expected that they cannot be learned from positive evidence alone. Nonetheless, the results described here show that, given sufficient positive evidence, these constraints (or rather, approximations to these constraints) are learnable from positive evidence. For suppose that the learner is never able to master these constraints. Then, the learner may persistently fail to realize that viable structures (such as 44a and 44b) are in fact allowed. This will lead to ineliminable on-going prediction errors: after *John is too stubborn to* …the learner will not consider that the sentence might continue with …*talk to*, or …*expect anyone to talk to*. Alternatively, the learner may falsely believe that nonviable structures (such as 44c and 44d) are part of the language. Thus, on hearing *John is too stubborn to*…, the learner may wrongly predict that the speaker may continue…*visit anyone who talked to* or…*ask anyone who talked to*. As we have noted, any ineliminable prediction errors, summed over predictions over an indefinitely large corpus, will lead our error measure to go to infinity. This is what the Prediction Theorem rules out: an ideal learner, with sufficient positive evidence, will learn to respect these linguistic constraints. This does not, of course, imply that the learner will necessarily respect these constraints by discovering the specific principles of the theory of government and binding; the theorem concerns the predictions of the learner, rather than the specific representational methods that the learner might use. This linguistic application suggests that the ability to learn to predict over a corpus requires finding all the linguistic regularities in that corpus. Thus, the ideal learner might be viewed as an "ideal structural linguist" (Harris, 1951), in that it finds the regularities in a language purely from exposure to a corpus of that language (although, of course, it merely outputs its predictions—it does not output a "theory" of the linguistic structure of the language, which is of course the goal of the linguist).

Chomsky (1965, 1957) has, however, re-oriented linguistics, to be concerned primarily with linguistic judgments,

rather than with attempts to find regularities in corpora. Most notably, speaker/hearers' judgments of which linguistic forms (phonological, syntactic, semantic) are acceptable in the language, are the primary linguistic data of linguistic theory. Human language acquisition clearly results in our ability to make such judgments—speakers of English typically agree that 44a and 44b are acceptable, and that 44c and 44d are not. Can the ability to make such judgments be learned purely from a corpus? Our analysis of grammaticality judgments, described above, indicates a positive result. An ideal learner will, with arbitrarily high probability, learn to be able to make approximately correct grammaticality judgments concerning stimuli of this kind, where the expected approximation becomes arbitrarily accurate, depending on the amount of available data.

The same point applies, more generally, to the wide range of linguistic phenomena that have been argued to be difficult or impossible to learn from positive evidence alone. For example, a well-known textbook, Crain and Lillo-Martin (1999) makes frequent use of the argument that constraints on what sentences can occur cannot be learned, and hence must be innate, because constraints can only be learned from negative data (data concerning what the constraints rule out). For example, they discuss the Empty Category Principle (ECP), the statement of which is rather technical, but which aims to explain patterns such as

(a) Who do you think Sarah will hire.

(b) Who do you think that Sarah will hire.

(c) Who do you think will win.

(d) *WHO do you think that will win. (45)

They argue "Like other constraints, the ECP is used to rule out ungrammatical sentences; hence it must be innate…" (p. 225). But according to the analysis above, this argument is not correct. Constraints *are* learnable from positive data alone; but, of course, the question of how much data is required to learn specific constraints, and whether the child is able to exploit this data, remains unresolved.

6.2. *The logical problem reconsidered: relationship to identification in the limit*

We noted at the outset that the scope for learning language from positive evidence alone has been viewed as limited in the light of Gold's (1967) classic paper "Language identification in the limit." These results were one motivation for the view that there is a fundamental logical problem with language acquisition from positive data. By contrast, the present results suggest that under very general conditions positive evidence can provide enough information for a learner to gain a great deal of information about a language (though we shall mention a number of caveats below).

Gold's (1967) paper, and the subsequent literature, has proved a range of positive and negative results concerning what can be learned from positive evidence. Gold's most celebrated result, and variants upon it, cast the problem of learning from positive evidence in what appears to be a negative light. Specifically, we define identification in the limit to require eventually correctly identifying a language (purely extensionally—i.e., picking out the, typically infinite, set of sentences that it does contain), from any text of that language (where a text is a semi-infinite sequence, i.e., with a determinate start, but no end item, of sentences of the language, such that each sentence in the language eventually appears). The learner need merely settle on the correct hypothesis and "stick" with it; it is not required that the learner is able to announce that it has identified the language successfully (and indeed this will typically not be possible). Now, we can informally state Gold's key result as follows: for any family of languages consisting of all finite languages (i.e., languages consisting of any finite set of sentences) and at least one infinite language, then that family of languages is not learnable in this limit. This means that there is at least one language, and a text generated by that language, such that the learner will not settle on the correct language, and stick with it, however much of the text it sees. The emphasis on finite languages is not crucial—similar negative results hold when learning only infinite languages (e.g., Niyogi, 2006). In particular, these results lead to the conclusion that finite state languages, and all languages generated by more complex grammatical formalisms are not learnable in the limit. Interestingly, these negative results have been extended to the case where the goal is merely probabilistically approximately correct (Valiant, 1984) identification of the target language (Niyogi, 2006), which follow, roughly, because almost any interesting class of languages has an infinite VC dimension (Vapnik, 1998; see Niyogi, 2006, for analysis and discussion).

The present results do not, of course, cast doubt on the validity of these negative results. Nor does it cast doubt on the usefulness of Gold's approach to the study of learning. Indeed, an important subfield of research, learning theory, has emerged from extensions of Gold's results (Angluin, 1980; Blum & Blum, 1975; Jain, Osherson, Royer, & Kumar Sharma, 1999; Martin & Osherson, 1998; Osherson, Stob, & Weinstein, 1985). Moreover, results from learning theory have been extensively related to human learning, including language learning (e.g., Niyogi, 2006; Osherson & Weinstein, 1982; Osherson, Stob, & Weinstein, 1982, 1984; Pinker, 1979, 1984).

The present results do emphasize the general truism that different formal idealizations of a single process—here the process of language acquisition—can lead to very different theoretical conclusions. The pressing question, therefore, is in what ways do the idealizations differ, and which idealization appears to be most relevant to how children learn natural language. An exhaustive analysis of the issues is beyond the scope of this paper. Here we briefly mention three critical points of difference (see Rohde & Plaut, 1999 for related discussion).

6.3. Identifying vs. modeling the language

A first difference is that Gold's criterion for successful learning is more exacting than that considered here. Gold is concerned with precisely 'identifying' a language—i.e., specifying exactly (or almost exactly—see Osherson et al., 1985) what sentences it does or does not contain. This seems too strict a criterion of learning in relation to how children learn language—after all the idiolects of any two native speakers will presumably show at least subtle differences. Moreover, even a single difference over a specific grammatical rule between two idiolects can lead two speakers to disagree on the grammaticality of the infinite number of sentences in which that grammatical rule is involved. Thus, we would expect that any two people would disagree on the grammaticality of an infinite number of sentences. This means that theoretical results showing that a learner cannot precisely identify a language from a teacher providing only positive evidence, such as Gold provides, may not apply directly to language acquisition in the child. The model developed here allows that the learner and the 'teachers' from whom the language is learned may make different judgments about the grammaticality of an infinite number of sentences (and the teachers may, presumably, also differ among themselves). But the learner and teachers will agree on almost all sentences that have a substantial probability of being said. This means that, for example, a disagreement between learner and teachers concerning the application of a controversial grammatical rule in a ten billion word long sentence will not count noticeably against the learner's having successfully acquired the language. From the pragmatic point of view of explaining how learners come to understand the actual sentences that they hear, and learn to produce similar sentences, the more relaxed criterion adopted here seems reasonable, and indeed, arguably required to explain "endogenous" aspects of language change (e.g., Niyogi, 2006).

6.4. The impact of statistical properties of language

A second, and related, difference is that Gold's result makes a crucial simplification in ignoring *statistical* properties of the language. In Gold's learning set-up, a language is a collection of sentences; and the goal of learning is to identify this set. But in the speech to which children are exposed, some types of sentences are more common than others—and learning the language critically involves learning these types of sentences, over and above types of sentences which are rarely or never produced. Thus, all native speakers of English agree that *the cat is on the mat* is an acceptable grammatical sentence; but examples of a rare structure, such as the multiply center-embedded such as *the cat the dog the man saw chased ran* leaves native speakers uncertain regarding grammaticality.

6.5. Worst case vs. typical case analysis

A third difference between Gold's framework and the present set-up is that Gold's original results demand that for a language to be learnable, it must be possible for the learner to learn the language given any *text* for that language. Here a text is defined as a (typically infinite) sequence of sentences (allowing arbitrary repetitions) which includes all and every sentence of the language. This means that every grammatical sentence of the language will be encountered eventually, but that there are typically no further constraints concerning the order in which sentences are encountered. Gold (1967) notes that the demand that language can be learned from every text may be too strong. That is, he allows the possibility that language learning from positive evidence may be possible precisely because there are restrictions on which texts are possible. As we have noted, when texts are restricted severely, e.g., they are independent, identical samples from a probability distribution over sentences, positive results become provable (e.g., Pitt, 1989); but the present framework does not require such restrictive assumptions.

6.6. The power of absence as implicit negative evidence

Indeed, once the demand that the learner must successfully acquire the language from any text is abandoned, then a potentially powerful source of 'implicit' negative evidence becomes available: absence as implicit negative evidence. To see how critically important this factor can be consider a language learner that is considering the viability of the 'vacuous' grammar, that any set of words in any order is grammatical—'anything goes.' But suppose that the ten million words that the learner has so far encountered have been generated by a trivial finite state grammar. It might seem that the learner can pretty safely rule out the 'vacuous' hypothesis, under these conditions—and, indeed, it might seem that any intelligent learning mechanism is likely to reach this conclusion. The *absence* of all but a tiny fraction of possible sentences would seem to be strong evidence that these sentences (or at least, the vast bulk of them) are *not* allowed in the language. Thus, it seems reasonable to interpret absence as a potential source of implicit negative evidence. But in Gold's set-up, a learner that adopts this assumption will be found wanting, because learners are required to acquire the language successfully, *whatever* the text on which they learn (so long as the text includes all and only the grammatical sentences of the language). Thus, any text at all is a perfectly legitimate text for the 'vacuous' grammar, including the one mentioned above; that is, the text can be 'rigged' arbitrarily to 'mislead' the learner; and Gold's criterion requires that the learner should, nonetheless, always ultimately succeed in identifying the language correctly. More broadly, because the text can be rigged arbitrarily, the learner can never rule out 'over-general' grammars—i.e., grammars that allow more sentences in the language than the target grammar.

Intuitively, the point is that for any 'reasonable' text, including the linguistic inputs to which children are exposed, absence can be used as negative evidence. Thus, by allowing 'unreasonable' texts, Gold's idealization makes the learning problem unduly difficult.

The potential importance of absence as a source of negative evidence applies not just at the general level mentioned above. As Rohde and Plaut (1999) have elegantly argued, it is also at the core of a wide range of specific proposals that attempt to explain how the child can acquire aspects of language from positive evidence alone. These proposals, which include the "uniqueness principle," "competition," "preemption," "blocking," the "principle of contrast," "mutual exclusivity" and the "M-constraint" (Bowerman, 1988; MacWhinney, 1993, 2004; Pinker, 1984; Wexler & Culicover, 1980), all rely on absence as an implicit signal than certain forms cannot occur. Rohde and Plaut (1999) point out that these principles require the learner to use 'soft' constraints such as that verbs typically have a single past tense, or that nouns typically have a single plural form. The constraints are 'soft' because they are some cases in which they do not apply. For example, in US English, 'dive' has two past tense forms 'dived' and 'dove,' both of which are reasonably frequent. But the soft constraint can nonetheless be extremely useful to the learner, if combined with the use of absence as negative evidence. Suppose, for example, that the learner hears countless examples of 'went' as the past tense of 'go.' The constraint that verbs typically have just one past tense means that the learner may reasonably conjecture that 'goed' is not viable. By using absence as surrogate negative evidence, the more examples of 'went' are heard, the more confident the learner can be. The learner can reason that if 'goed' existed, it would very likely have been encountered. Indeed, presumably it is just such an inference which underlies our adult intuition that 'goed' is not viable—it would seem incredibly unlikely that 'goed' is a valid past tense form, but that due to a remarkable chain of coincidence, one has never heard anyone say it. Note, by contrast, that this style of reasoning would not be appropriate in the context of a typical learnability set-up; this is because the learner must succeed even in the 'rigged' text where 'goed' is legitimate, but is only heard after one billion examples of 'went.'

To use absence as a source of negative evidence requires, then, some restrictions on the class of possible inputs to the learner (texts cannot be arbitrarily rigged). But which assumptions about the class of texts are appropriate? One extreme idealization would be to assume that texts are created by concatenating sentences chosen independently from an identical distribution over the (infinite) space of possible sentences (e.g., Horning, 1969). This idealization is attractive from a formal point of view—because it allows the application of the standard probability theory concerned with the properties of such sequences. But this assumption is clearly too restrictive, because there are patently very strong, and linguistically crucial, interdependencies between successive sentences. A natural direction to explore is to weaken this assumption by allowing dependencies between short sub-sequences of sentences, or in some other way assume that the language is relatively stationary (Rohde & Plaut, 1999).

Any such assumption that the language is 'stationary' is subject to the concern, however, that there are dependencies between chunks of language over arbitrary scales. To see this, consider the dependencies in an academic journal, which apply between sentences and subsequent sentences; between paragraphs and subsequent paragraphs; between sections and subsequent sections; and even between articles and subsequent articles. Thus, it is not clear that language is a stationary stochastic process over any time-scale, although the possibility remains that it may be approximately stationary, to some useful degree, or at some level of linguistic analysis. The present framework places strong, but rather general, restrictions on texts, but without requiring stationarity. Specifically, infinite texts must be monotone computable. This restriction is significant. The overwhelming majority of infinite texts will correspond to uncomputable sequences.[28] However, the uncomputable sequences, being incompressible (every initial segment is incompressible) to some degree, correspond more or less to "white noise" and have no meaning or regularity, and hence there is no cogent reason why one should want to learn them or that they would express any interesting structure. Note, too that the restriction to computability is still quite weak, in the sense that it does not impose any constraints which are specific to learning natural language. Nonetheless, the results we have discussed here show that adding this constraint on inputs suffices to make language learning possible.

7. Summary

In a nutshell, Gold's learning paradigm embodies the view that the child's goal in learning language is primarily *theoretical*: the goal is to get the correct *theory* that decides all possible cases, whether or not they arise in practice or not; and Gold demands that this theory is learnable on all possible texts for the language. But it may be more appropriate to view the child's primary goal as *practical*: What matters is learning to handle the language as it is actually spoken, from samples of the language that might actually be heard. In brief, Gold's results show that language learning from positive evidence alone is impossible, when viewed as a problem of theory discovery; but the present results show that practical knowledge of how to predict, judge and produce sentences of a language can in principle be derived from positive evidence alone. The present analysis seems appropriate for natural languages where there is typically little consensus concerning what

[28]This follows because the number of computable texts is bounded by the number of Turing machines, which is countable; but the set of all infinite texts is uncountable.

constitute correct sentences is necessarily fluid: different native speakers and linguists may completely disagree on the correctness of infinitely many sentences; and grammaticality judgments may be inconsistent across different occasions for the same speaker.

7.1. Open questions

The analysis in this paper considers the amount of information available to a learner from positive evidence alone; but it does not consider the extent to which it is possible for a learner to exploit this information fully.

To consider whether this information can be exploited fully, let us assume that the learner has the same computational power as the mechanism producing the corpus (this seems a reasonable assumption, as today's learner is tomorrow's corpus-generator for future learners). Thus, the learner is modeled as a monotone Turing machine with access to a random input. To obtain optimal learning, the learner needs to predict according to the universal distribution, λ, conditional on previous input. But in general, at least, this will not be possible, because λ is an uncomputable distribution—this is a standard result of Kolmogorov complexity theory (Li & Vitányi, 1997). So, although the information may be available, the learner cannot exploit it fully.

Hence, a psychological mechanism that learns using a Simplicity Principle must operate by *approximating* the probability distribution λ—i.e., finding a short, but not necessarily the shortest, encoding of past linguistic data. This opens up the very interesting question of how approximations to λ will fare in language acquisition—in prediction, making grammaticality judgments, and language production. Two extreme possibilities may be envisaged. One extreme possibility is that computational restrictions change the picture dramatically. Although for a learner with no computational limitations, the linguistic input contains enough information for successful learning, it might be that for real computational learners, very little useful information about language structure can be extracted from the input. The other extreme possibility is that computational limitations do not qualitatively affect what can be learned—i.e., the learner can predict, judge grammaticality, and produce language successfully, by choosing the simplest account of the language that it is able to find, although not, of course, quite as accurately as would be possible if the Simplicity Principle could be implemented precisely. The question of which extreme represents the true situation, or which compromise between them is appropriate, is currently an open problem. Nonetheless, some steps have been made in this direction. Vitányi and Li (2000) consider a computable approximation to the universal distribution—the statistical Minimum Description Length Principle (e.g., Rissanen, 1987, 1989) and show via mathematical analysis that, under certain conditions, this computable approximation is expected to lead to successful predictions with probability 1. There remains, though, a rich set of open questions concerning the properties of learners with various more specific computational properties and restrictions (e.g., learners that can only entertain certain languages). Most important, of course, is the analysis of idealized learners that are psychologically realistic as models of human learners.

A related area set of questions concerns more specific models of both of the language to be learned, and of the nature of the learner. In the analysis here, our only constraint on the language is that it could be produced by a 'monotone' Turing computable process (with access to a source of randomness). The learning problem may be expected to become substantially easier if constraints are placed in the class of languages that might need to be learned. These constraints might range from very general properties of language (which might emerge from communicative constraints, cognitive limitations, or in a variety of other ways) to highly specific and elaborate constraints of language structure, such as those embodied in 'universal grammar' (Chomsky, 1981).

A third important set of open questions, that we touched on at the end of the last subsection, concerns the quality and amount of data required for language acquisition to occur. Formal results both in the tradition of formal learning theory started by Gold and learning by simplicity started by Solomonoff have focussed on learning in the asymptote, using a potentially infinite supply to data. But real language learning must occur reliably using limited amounts of data (although the available data to the child will comprise many millions of words each year). Thus a crucial set of open questions concerns how rapidly learners can converge well enough on the structure of the linguistic environment to succeed reasonably well in prediction, grammaticality judgments and language production. Some progress on this issue has already been made by Solomonoff (1978), who has shown that the expected squared error in the nth prediction probabilities of using the universal distribution to decrease more rapidly than $1/(n \log(n))$ (see Li & Vitányi, 1997).

8. Conclusion

This paper presents some positive results concerning what is learnable from positive linguistic data. We have seen that exposure to positive data is sufficient for an ideal learner to predict new material from a corpus, learn to make grammaticality judgments, and learn to produce language. These results re-open the question of the viability of language learning from positive evidence under less idealized conditions, of limited computational resources or amounts of linguistic data available to the learner. The framework developed here presents a complementary idealization of the problem of language acquisition to that initiated by Gold (1967). They also suggest that purely logical arguments against the general viability of language acquisition from positive evidence may need to be rethought.

Acknowledgments

We would like to thank James Adelman, Morten Christiansen, Mark Ellison, Steven Finch, Markus Gonitizke, Ulrike Hahn, Padraic Monaghan, Marjolein Merkx, Stephen Muggleton, Gregory Mulhauser, Mike Oaksford, Luca Onnis, Martin Pickering, Emmanuel Pothos, Matthew Roberts, Jerry Seligman and Gerry Wolff and two anonymous reviewers for valuable discussions of these ideas at various stages in their development or for comments on this manuscript. Nick Chater has been supported by a Senior Research Fellowship from the Leverhulme Trust. Nick Chater was supported in part by European Commission grant RTN-HPRN-CT-1999-00065, by the ESRC, the DTI, the Human Frontiers Science Program, Mercer Oliver Wyman, and Decision Technology. Paul Vitányi was supported by the European Commission Neurocolt II WG EP 27150 and the Quantum Algorithms and Information Processing Project IST-1999-11234, and the ESA Program. Thanks to Ursula Richards and Emma Johnstone for helping chasing down the references and putting the manuscript in order.

Appendix A. Measures and semi-measures

We want to specify a probability distribution over one-way binary sequences of 0 s and 1 s. This requires us to introduce the notions of measures and semi-measures. A non-standard approach to measures is standard in the relevant areas of Kolmogorov complexity theory, namely the sub-field of algorithmic probability (see Li & Vitányi, 1997, pp. 242–244).

Let us define a probability *measure,* ϕ, over these sequences, as satisfying:

$$\phi(\text{empty-string}) = 1, \tag{A.1}$$

$$\phi(x) = \phi(x0) + \phi(x1), \tag{A.2}$$

where x is a binary sequence. According to this definition, the empty string has probability 1; for any n, the sum of the probabilities of strings of length n is also 1.

For the analysis below, it is convenient to introduce a more general notion. Let us define a probability *semi-measure*, σ, over these sequences, as satisfying:

$$\sigma(\text{empty-string}) \leqslant 1, \tag{A.3}$$

$$\sigma(x) \geqslant \sigma(x0) + \sigma(x1). \tag{A.4}$$

According to this definition, the probability of the empty string is less than 1. The sum of the probability of strings of length $n+1$ will be less than or equal to the sum of the probability of the strings of length n. The definitions of measures and semi-measures can be generalized in the obvious ways to sequences which can contain more than two symbols.

Notice that monotone computable distributions, μ, discussed above will typically be semi-measures, but not measures. This is because there may be some inputs that lead to undefined outputs from the associated monotone machine at some point in the output sequence, so that the sum probability over the output sequences will be less than 1. In particular, the universal monotone distribution, λ, is a semi-measure rather than a measure.

Appendix B. Proof of Step 3

We need to prove

$$D(\mu||\lambda) = \sum_{j=1}^{\infty} D_j(\mu||\lambda), \tag{B.1}$$

where μ is a measure and λ is a semimeasure.

$$\begin{aligned}
D(\mu||\lambda) &= \sum_{x_1\ldots x_j\ldots} \mu(x_1\ldots x_j\ldots)\log\frac{\mu(x_1\ldots x_j\ldots)}{\lambda(x_1\ldots x_j\ldots)} \\
&= \sum_{x_1\ldots x_j\ldots} \mu(x_1\ldots x_j\ldots) \\
&\quad \times \log\left(\frac{\mu(x_1)}{\lambda(x_1)}\cdot\frac{\mu(x_2|x_1)}{\lambda(x_2|x_1)}\cdots\frac{\mu(x_j|x_1\ldots x_{j-1})}{\lambda(x_j|x_1\ldots x_{j-1})}\cdots\right) \\
&= \sum_{x_1\ldots x_j\ldots} \mu(x_1\ldots x_j\ldots)\log\frac{\mu(x_1)}{\lambda(x_1)} \\
&\quad + \sum_{x_1\ldots x_j\ldots} \mu(x_1\ldots x_j\ldots)\log\frac{\mu(x_2|x_1)}{\lambda(x_2|x_1)} + \cdots \\
&\quad + \sum_{x_1\ldots x_j\ldots} \mu(x_1\ldots x_j\ldots)\log\frac{\mu(x_j|x_1\ldots x_{j-1})}{\lambda(x_j|x_1\ldots x_{j-1})} + \cdots \\
&= \sum_{x_1}\mu(x_1)\log\frac{\mu(x_1)}{\lambda(x_1)}\sum_{x_2\ldots x_j\ldots}\mu(x_2\ldots x_j\ldots|x_1) \\
&\quad + \sum_{x_1,x_2}\mu(x_1,x_2)\log\frac{\mu(x_2|x_1)}{\lambda(x_2|x_1)} \\
&\quad \times \sum_{x_3\ldots x_j\ldots}\mu(x_3\ldots x_j\ldots|x_1,x_2)\ldots \\
&\quad + \sum_{x_1\ldots x_j}\mu(x_1\ldots x_j)\log\frac{\mu(x_j|x_1\ldots x_{j-1})}{\lambda(x_j|x_1\ldots x_{j-1})} \\
&\quad \times \sum_{x_{j+1}\ldots}\mu(x_{j+1}\ldots|x_1\ldots x_j) + \cdots.
\end{aligned} \tag{B.2}$$

Because μ is a measure, the sum of conditional probabilities of all possible continuations of a sequence is 1. This means that, for any j,

$$\sum_{x_{j+1}\ldots}\mu(x_{j+1}\ldots|x_1\ldots x_j) = 1, \tag{B.3}$$

leading to the simplified form of (B.2) as

$$\begin{aligned}
&= \sum_{x_1}\mu(x_1)\log\frac{\mu(x_1)}{\lambda(x_1)} + \sum_{x_1,x_2}\mu(x_1,x_2)\frac{\mu(x_2|x_1)}{\lambda(x_2|x_1)}\cdots \\
&\quad + \sum_{x_1\ldots x_j}\mu(x_1\ldots x_j)\frac{\mu(x_j|x_1\ldots x_{j-1})}{\lambda(x_j|x_1\ldots x_{j-1})} + \cdots.
\end{aligned} \tag{B.4}$$

This can be rewritten as

$$= \sum_{x_1} \mu(x_1) \log \frac{\mu(x_1)}{\lambda(x_1)} + \sum_{x_1} \mu(x_1) \sum_{x_2} \mu(x_2|x_1) \frac{\mu(x_2|x_1)}{\lambda(x_2|x_1)} \cdots + \sum_{x_1 \ldots x_{j-1}} \mu(x_1 \ldots x_{j-1}) \sum_{x_j} \mu(x_j|x_1 \ldots x_{j-1}) \times \frac{\mu(x_j|x_1 \ldots x_{j-1})}{\lambda(x_j|x_1 \ldots x_{j-1})} + \cdots, \quad \text{(B.5)}$$

which by the definition of standard Kullback–Liebler distance (Eqs. (17) and (19)), can be rewritten as

$$= D(\mu(x_1)||\lambda(x_1)) + \sum_{x_1} \mu(x_1) D(\mu(x_2|x_1)||\lambda(x_2|x_1)) \cdots + \sum_{x_1 \ldots x_{j-1}} \mu(x_1 \ldots x_{j-1}) \times D(\mu(x_j|x_1 \ldots x_{j-1})||\lambda(x_j|x_1 \ldots x_{j-1})) + \cdots, \quad \text{(B.6)}$$

which by the definition of the Kullback–Liebler distance, D_j, (Eq. (20)) is

$$= \sum_{j=1}^{\infty} D_j(\mu||\lambda). \quad \text{(B.7)}$$

This proves the theorem.

There is, however, a complication, with respect to applying this theorem in the wider context of the proof. This is because the proof holds only if μ is a measure, whereas, in general μ will often be a semi-measure. This is because μ may can correspond to an arbitrary monotone Turing machine M, and such machines will not typically lead to a well-defined output for each input (e.g., the machine may halt or go into an infinite loop after a certain input, and then produce no further output). Given that this can happen the sum of the probabilities over all possible infinite output sequences of 0 and 1 s will be less than 1, because of the non-zero probability than a well-defined infinite output will not be produced at all. Li and Vitányi (1997) show that this can be handled simply by adding a third symbol 'u' for undefined. In these terms, an input that leads to the output 010100 and then goes into an infinite loop is viewed as producing the infinite sequence 010100$uuu\ldots$ Thus, we define μ (and hence also λ) as a measure over the set of infinite sequences of the three symbols 0, 1 and u. This means that the Kullback–Liebler divergences above are defined over sequences of the three symbols. We shall pick up the ramifications of this complication for Step 4 of the proof in Appendix C.

Appendix C. Proof of part of Step 4

The material in this section is an elaboration of the discussion in Li and Vitányi (1997, p. 329). We have to show that

$$(P(0) - Q(0))^2 \leqslant \frac{\log_e 2}{2} D(P||Q), \quad \text{(C.1)}$$

where P and Q range over 0 and 1. Define $p = P(0)$ and $q = Q(0)$. Then we expand the Kullback–Liebler term, and rewrite the result in terms of p and q to give the inequality:

$$(p-q)^2 \leqslant \frac{\log_e 2}{2} \left(p \log_2 \frac{p}{q} + (1-p) \log_2 \frac{1-p}{1-q} \right) = \frac{1}{2} \left(p \log_e \frac{p}{q} + (1-p) \log_e \frac{1-p}{1-q} \right), \quad \text{(C.2)}$$

where the equality involves switching from base 2 to base e, which will be convenient below, using the fact that

$$\log_e 2 \cdot \log_2 x = \log_e x. \quad \text{(C.3)}$$

The inequality will hold if, for all p and q:

$$p \ln \frac{p}{q} + (1-p) \ln \frac{1-p}{1-q} - 2(p-q)^2 \geqslant 0, \quad \text{(C.4)}$$

let us fix p and treat it as a constant. We can then consider the left hand side of the inequality (C4) as a function of q, $F(q)$. To prove the inequality, we need to show that $F(q) \geqslant 0$, whatever the value of q, for arbitrary p.

It is useful to consider in which direction $F(q)$ changes as q changes—i.e., to know the derivation dF/dq, which can evaluate term by term:

$$\frac{dF}{dq} = p/q + (1-p)/(1-q) + 4(p-q) \quad \text{(C.5)}$$

and rearrange:

$$= \frac{1}{q(1-q)} (q - p + 4(p-q)q(1-q)). \quad \text{(C.6)}$$

Because $1/q(1-q)$ is positive, $(dF/dq) \geqslant 0$ if and only if

$$q - p + 4(p-q)q(1-q) \geqslant 0, \quad \text{(C.7)}$$

which can be rewritten:

$$(q-p)(1 - 4q(1-q)) \geqslant 0. \quad \text{(C.8)}$$

Note that $1 - 4q(1-q) \geqslant 0$ for any q between 0 and 1—specifically, the left hand side is always nonnegative, reaching a minimum of 0, when q is 1/2. This means that the sign of dF/dq depends only on the $q-p$ term. This means that: if $q > p$, $(dF/dq) \geqslant 0$, and if $q < p$, $(dF/dq) \leqslant 0$.

We are now in a position to show that $F(q) \geqslant 0$ for all q. First, note that if $q = p$, we have

$$F(p) = p \ln \frac{p}{p} + (1-p) \ln \frac{1-p}{1-p} - 2(p-p)^2 = 0. \quad \text{(C.9)}$$

Now suppose that we increase q above p. Where $q > p$, $(dF/dq) \geqslant 0$, which implies that $F(q)$ will increase, and hence that $F(q) \geqslant 0$. Suppose instead that we decrease q below p. Where $q < p$, $(dF/dq) \leqslant 0$, which implies that $F(q)$ will increase as q decreases, and hence again that $F(q) \geqslant 0$. Thus, we have shown that $F(q) \geqslant 0$ for all q, and for arbitrary p, and the theorem is proved.

A final complication arises because, as we noted in the proof of Step 3 (see the discussion at the end of Appendix B), the Kullback–Liebler distance between μ and λ is calculated over three symbols, 0, 1, u—we shall write this $D^{0,1,u}(\mu,\lambda)$. But the result bounding sum-squared error above applies only for the case where there are two symbols. To fill in this gap in the proof, we consider the Kullback–Liebler distance between μ and λ for two symbols, 0 and v, where v collapses together 1 and u. This distance, which we shall write $D^{0,v}(\mu,\lambda)$ is binary, and hence the upper bound on sum-squared error applies. Moreover, we shall show that Kullback–Liebler distance can only decrease when symbols are collapsed in this way—and hence that the Kullback–Liebler distance between μ and λ calculated over the three symbols, 0, 1, u must exceed the two symbol case, which itself must exceed sum-squared error. Thus the three-symbol Kullback–Liebler distance does provide an upper bound on sum-squared error.

To complete the proof, then, we need to show that $D^{0,1,u}(\mu,\lambda) \geqslant D^{0,v}(\mu,\lambda)$. To show this, we start by rewriting the three outcomes of $D^{0,1,u}(\mu,\lambda)$ as sequences: 00, $v1$, vu, where we define $\mu_{\text{seq}}(00) = \mu(0)$, $\mu_{\text{seq}}(v1) = \mu(1)$ and $\mu_{\text{seq}}(vu) = \mu(u)$, and similarly for λ. All other two item sequences have zero probability. The Kullback–Liebler divergence between μ and λ over these two symbol sequences will be just $D^{0,1,u}(\mu,\lambda)$, because there are three outcomes with exactly the same probabilities as in the standard definition of $D^{0,1,u}(\mu,\lambda)$. But using the representation as sequences of two items, we can use a general result, which is a special case of main derivation in Step 3 above:

$$\begin{aligned} D^{0,1,u}(\mu,\lambda) &= D(\mu_{\text{seq}}(x_1x_2)||\lambda_{\text{seq}}(x_1x_2)) \\ &= \sum_{x_1=0,v;x_2=0,1,u} \mu_{\text{seq}}(x_1x_2)\log\frac{\mu_{\text{seq}}(x_1x_2)}{\lambda_{\text{seq}}(x_1x_2)} \\ &= \sum_{x_1=0,v}\mu(x_1)\log\frac{\mu(x_1)}{\lambda(x_1)} + \sum_{x_1=0,v}\mu(x_1) \\ &\quad\times\sum_{x_2=0,1,u}\mu_{\text{seq}}(x_2|x_1)\log\frac{\mu_{\text{seq}}(x_2|x_1)}{\lambda_{\text{seq}}(x_2|x_1)} \\ &= D^{0,v}(\mu(x_1)||\lambda(x_1)) \\ &\quad+\sum_{x_1}\mu(x_1)D(\mu_{\text{seq}}(x_2|x_1)||\lambda_{\text{seq}}(x_2|x_1)), \end{aligned} \tag{C.10}$$

Kullback–Liebler distance cannot be negative, as we have noted, so the sum on the right hand side cannot be negative, and we can conclude:

$$D^{0,1,u}(\mu,\lambda) \leqslant D^{0,v}(\mu||\lambda), \tag{C.11}$$

which proves the result. □

Appendix D. Proof of the re-scaling lemma

We have a probability distribution $P(y_i)$. We wish to encode these outcomes according to a probability distribution $Q(y_i)$. A proper subset of outcomes that Q assigns a non-zero probability are actually impossible in P (i.e., $Q(y_i) > 0$ but $P(y_i) = 0$). The re-scaling lemma states that given this constraint the minimum expected number of bits of information wasted, measured by the Kullback–Liebler divergence, $D(P\|Q)$, is $\log_2(1/(1-\Delta_j))$, which is attained when, for all $y_j \notin S_0$,

$$Q(y_j) = (1-\Delta)P(y_j). \tag{D.1}$$

That is, the minimum waste is obtained by *re-scaling* all the $P(y_i)$ values that can occur into the available probability for these items under the distribution Q.

Proof. Consider a probability distribution P. We want to find the probability distribution $Q(y_i)$ which minimizes the Kullback–Liebler divergence $D(P\|Q)$, subject to the constraint that there is a subset S_0 of outcomes y_i for which $Q(y_i) > 0$, but $P(y_i) = 0$, such that

$$\sum_{y_j \in S_0} Q(y_j) = \Delta. \tag{D.2}$$

The Kullback–Liebler distance between P and Q will be the same as the Kullback–Liebler distance between probability distributions over *sequences* with the same probabilities (see the last part of Appendix C for a similar method). Define $Q_{\text{seq}}(0y_i) = Q(y_i)$ for y_i in S_0; and $Q_{\text{seq}}(1y_i) = Q(y_i)$ for y_i not in S_0. These are the only allowable sequences—other binary sequences have probability 0. Define $P_{\text{seq}}(0y_i) = 0$ and $P_{\text{seq}}(1y_i) = P(y_i)$ for all y_i. It is easy to verify that the probabilities of the different sequences are the same as the probabilities of the single outcomes in the original distributions so that

$$D(P||Q) \equiv D(P_{\text{seq}}||Q_{\text{seq}}). \tag{D.3}$$

Now, using the representation as sequences of two items, we can use a standard result, which is a special case of the result used in the main derivation of Step 3 of the prediction theorem (Eq. (11)) above:

$$\begin{aligned} D(R(x_1x_2)||S(x_1x_2)) &= D(R(x_1)||S(x_1)) \\ &\quad+\sum_{x_1}R(x_1)D(R(x_2|x_1)||S(x_2|x_1)) \end{aligned} \tag{D.4}$$

and apply the special cases of P_{seq} and Q_{seq} we obtain

$$\begin{aligned} &D(P_{\text{seq}}(x_1x_2)||Q_{\text{seq}}(x_1x_2)) \\ &= P(1)\log_2\frac{P(1)}{Q(1)} + P(0)\log_2\frac{P(0)}{Q(0)} \\ &\quad+P(1)D(P_{\text{seq}}(y_j|1)||Q_{\text{seq}}(y_j|1)) \\ &\quad+P(0)D(P_{\text{seq}}(y_j|0)||Q_{\text{seq}}(y_j|0)). \end{aligned} \tag{D.5}$$

This can be simplified using the facts that $P(1) = 1$, $P(0) = 0$, $Q(1) = 1-\Delta$, $Q(0) = \Delta$, $P_{\text{seq}}(y_i|1) = P(y_i)$, $P_{\text{seq}}(y_i|0) = 0$. Moreover, we know that for y_i not in S_0, $Q_{\text{seq}}(y_i|1) = Q_{\text{seq}}(1\ y_i)/Q'(1) = Q(y_i)/(1-\Delta)$. The resulting simplification is

$$= \log_2\frac{1}{1-\Delta} + D\left(P(y_j)||\frac{Q(y_j)}{1-\Delta}\right). \tag{D.6}$$

The choice of Q only affects the second term. (D.6) can be minimized if the two distributions compared by Kullback–Liebler divergence are the same (we use the general result that Kullback–Liebler diverge is minimal, and attains 0, only between a probability distribution and itself). This means the minimum is attained when:

$$P(y_j) = \frac{Q(y_j)}{1-\Delta}. \tag{D.7}$$

When rearranged, this gives the required result. □

Appendix E. Proof of the undergeneralization theorem

Suppose that a past sequence of words, x, has been encountered. The next word, k, is allowed by the true grammar, and has a non-zero probability of being said, but is disallowed by the learner's probability distribution. As usual, we assume that language is generated from a monotone computable probability distribution, P_μ, over word sequences; and that the learner is using the universal prior distribution, P_λ, over word sequences. Then undergeneralization will occur when $P_\mu(xk)>0$, but when $P_\lambda(xk)=0$. It is convenient to convert this formulation into the equivalent binary representation. Suppose that the word sequence xk corresponds to the binary sequence y. Then, because the binary code stands in one-to-one correspondence with the representation in terms of word sequences, the criterion for undergeneralization can be stated as: $\mu(y)>0$, whereas $\lambda(y)=0$. Can there be such a sequence, y? There cannot, because by applying Eq. (14) above, we obtain

$$\lambda(y) \geqslant 2^{-Km(\mu)}\mu(y) > 0. \tag{E.1}$$

This implies, trivially, that $\mu(y)>0$, then $\lambda(y)>0$. Hence undergeneralizations cannot occur.

Appendix F. Proof of the soft undergeneralization theorem

Suppose that the sequence of words encountered by the learner is generated according to a computable probability distribution P_μ, and that the learner attempts to predict this sequence by a universal probability distribution P_λ. We denote the sequence of the initial $j-1$ words that the learner encounters by x, and let us call the jth word, k. If the learner undergeneralizes on word k by a factor f, this means that the learner underestimates the probability that k will occur after x by a factor f, i.e., $f\cdot P_\lambda(k|x) \leqslant P_\mu(k|x)$. We write $\Lambda_j(x)$ to denote the probability that the k that is chosen according to the true distribution is a word on which the learner undergeneralizes, given the preceding sequence, x. $\Lambda_j(x)$, can be expressed:

$$\Lambda_j(x) = \sum_{k:P_\lambda(k|x)f \leqslant P_\mu(k|x)} P_\mu(k|x). \tag{F.1}$$

The *expected* probability, $\langle\Lambda_j\rangle$, with which this occurs on the jth item is

$$\langle\Lambda_j\rangle = \sum_{x:l(x)=j-1} P_\mu(x)\Lambda_j(x). \tag{F.2}$$

The Soft Undergeneralization Theorem states that

$$\sum_{j=1}^{\infty}\langle\Lambda_j\rangle \leqslant Km(\mu)\,\frac{1}{\log_2 f/e} \tag{F.3}$$

(so long as $f>e$).

Proof. The overall logic of the proof is similar to that used in proving the overgeneralization theorem. As before, we consider the scenario in which the learner uses its probability distribution P_λ to *encode* the output from the true distribution, P_μ. After the sequence x, of $j-1$ words, has been encountered, we can ask: what is the expected amount of wasted information in encoding the jth item? As in the case of overgeneralization, such waste is inevitable, because the learner is using codes which are optimized to the learner's distribution (i.e., $P_\lambda(\cdot|x)$), rather than to the true (but from the learner's point of view, unknown) distribution (i.e., $P_\mu(\cdot|x)$). The key underlying intuition is that, to the extent that the learner has a tendency to undergeneralize, the learner must necessarily waste a certain amount of information. This is because the learner encodes some items with long codes, because the learner assumes that they are very unlikely; but in reality, they are likely, and hence should optimally be assigned short codes. By using long codes where short codes would do, the learner therefore wastes information in encoding the sequence. The greater to degree to which the learner undergeneralizes, the greater the amount of wasted information.

More specifically, the aim of the proof is to put a lower bound on the amount of information that is wasted (as measured by Kullback–Liebler divergence), given that a specified amount of undergeneralization occurs.

To get started, we aim to specify how information waste can be minimized, given that a certain amount of undergeneralization occurs. That is, suppose that, instead of the specific distributions, P_μ and P_λ, we consider arbitrary distributions Q and R (where Q stands in for P_μ, and is viewed as the true distribution, and R stands in for P_λ, and is viewed as the learner's distribution). The only constraint on R and Q, is that R undergeneralizes with respect to Q with probability, Λ. We then specify the distributions Q and R in a way that we can show minimizes the information wasted. This amount of waste incurred in this 'minimal' case must therefore be a lower bound on the amount of waste incurred in the case where we use the distributions of interest, P_μ and P_λ. In an analogous aspect of the proof of the overgeneralization theorem, the 're-scaling lemma' (Appendix D) showed that the lowest information loss was achieved by specifying the learner's distribution as a re-scaled version of the true distribution (for the items where overgeneralization did not occur). We shall see that a similar, though slightly more complex, result holds here.

Lemma. *Consider probability distributions Q and R over outcomes, i, with a probability, Λ, (with respect to the 'true' distribution Q) that an outcome i arises for which R 'undergeneralizes' with respect to Q by a factor of at least f. That is, we assume that*

$$\Lambda = \sum_{i \in U} Q(i), \tag{F.4}$$

where $U = \{i | R(i) f \leqslant Q(i)\}$. (The set U consists of the items on which undergeneralization occurs.) Then,

$$D(Q\|R) \geqslant \Lambda \log_2 \frac{f}{e} \tag{F.5}$$

as long as $f > e$. Thus, this lemma relates the amount of undergeneralization to the amount of informational waste involved in using R to encode Q.

We now prove the Lemma. We break down the proof into two steps. First, we specify that the sum probability in R of the items that are underestimated by R is Λ', where

$$\Lambda' = \sum_{i:R(i)f \leqslant Q(i)} R(i). \tag{F.6}$$

Now we consider how the probabilities for all the $R(i)$ should be set in order to minimize $D(Q\|R)$. The second step is to consider the optimal value of Λ' to minimize $D(Q\|R)$.

To prove the first step, we begin by rewriting the distribution R in a rather indirect way, in terms of a new distribution S, where $R(i) = (\Lambda'/\Lambda)S(i)$ for $i \in U$; and $R(i) = (1 - \Lambda'/1 - \Lambda)\, S(i)$ for $i \notin U$. Note that S sums to one, and hence is a probability distribution, as shown below:

$$\begin{aligned}\sum_i S(i) &= \sum_{i \in U} S(i) + \sum_{i \notin U} S(i) = \sum_{i \in U} \frac{\Lambda}{\Lambda'} R(i) + \sum_{i \notin U} \frac{1-\Lambda}{1-\Lambda'} R(i) \\ &= \frac{\Lambda}{\Lambda'} \sum_{i \in U} R(i) + \frac{1-\Lambda}{1-\Lambda'} \sum_{i \notin U} R(i) \\ &= \frac{\Lambda}{\Lambda'} \Lambda' + \frac{1-\Lambda}{1-\Lambda'}(1-\Lambda') = 1. \end{aligned} \tag{F.7}$$

Now we adopt the method used in proving the re-scaling lemma above (Appendix D). The Kullback–Liebler distance between Q and R will be the same as the Kullback–Liebler distance between probability distributions over *sequences* with the same probabilities. Let us call the distributions over sequences, corresponding to Q and R, Q_{seq} and R_{seq} respectively. Define $Q_{\text{seq}}(0i) = Q(i)$ for $i \in U$; and $Q_{\text{seq}}(1i) = Q(i)$ for $i \notin U$. Similarly, define $R_{\text{seq}}(0i) = R(i)$ for $i \in U$; and $R_{\text{seq}}(1i) = R(i)$ for $i \notin U$. These are the only allowable sequences—other sequences have probability 0 in both Q and R.

Now we write down probabilities associated with the sequential representation. The probabilities associated with the first symbol are: $Q_{\text{seq}}(0) = \sum_{i \in U} Q(i) = \Lambda$; $Q_{\text{seq}}(1) = 1 - \Lambda$; $R_{\text{seq}}(0) = \sum_{i \in U} R(i) = \Lambda'$; $R_{\text{seq}}(1) = 1 - \Lambda'$. By routine calculation, the conditional probabilities of the second symbol, given the first symbol, are: $Q_{\text{seq}}(i|0) = Q(i)/\Lambda$; $Q_{\text{seq}}(i|1) = Q(i)/1 - \Lambda$; $R_{\text{seq}}(i|0) = R(i)/\Lambda'$; $R_{\text{seq}}(i|1) = R(i)/\ 1 - \Lambda'$. For these last two expressions, we substitute S for R, to obtain $R_{\text{seq}}(0i) = (1/\Lambda')(\Lambda'/\Lambda)S(i) = S(i)/\Lambda$ and $R_{\text{seq}}(1i) = 1/1 - \Lambda'((1-\Lambda')/(1-\Lambda))S(i) = S(i)/1 - \Lambda$.

Now the Kullback–Liebler distance between the *sequences* Q and R is defined as (adapting Eq. (D.4)):

$$\begin{aligned} &D(Q_{\text{seq}}(x_1x_2)||R_{\text{seq}}(x_1x_2)) \\ &= D(Q_{\text{seq}}(x_1)||R_{\text{seq}}(x_1)) \\ &\quad + \sum_{x_1} Q_{\text{seq}}(x_1) D(Q_{\text{seq}}(x_2|x_1)||R_{\text{seq}}(x_2|x_1)). \end{aligned} \tag{F.8}$$

Expanding and filling in the specific formulae above gives the following derivation:

$$\begin{aligned} D(Q||R) &= Q_{\text{seq}}(0) \log_2 \frac{Q_{\text{seq}}(0)}{R_{\text{seq}}(0)} \\ &\quad + Q_{\text{seq}}(1) \log_2 \frac{Q_{\text{seq}}(1)}{R_{\text{seq}}(1)} \\ &\quad + Q_{\text{seq}}(0) \left[\sum_{i \in U} Q_{\text{seq}}(i|0) \log_2 \frac{Q_{\text{seq}}(i|0)}{R_{\text{seq}}(i|0)} \right] \\ &\quad + Q_{\text{seq}}(1) \left[\sum_{i \notin U} Q_{\text{seq}}(i|1) \log_2 \frac{Q_{\text{seq}}(i|1)}{R_{\text{seq}}(i|1)} \right] \\ &= \Lambda \log_2 \frac{\Lambda}{\Lambda'} + (1-\Lambda) \log_2 \frac{1-\Lambda}{1-\Lambda'} \\ &\quad + \Lambda \left[\sum_{i \in U} (Q(i)/\Lambda) \log_2 \frac{Q(i)/\Lambda}{S(i)/\Lambda} \right] \\ &\quad + (1-\Lambda) \left[\sum_{i \notin U} (Q(i)/1-\Lambda) \log_2 \frac{Q(i)/1-\Lambda}{(S(i)/1-\Lambda)} \right] \\ &= \Lambda \log_2 \frac{\Lambda}{\Lambda'} + (1-\Lambda) \log_2 \frac{1-\Lambda}{1-\Lambda'} \\ &\quad + \sum_i Q(i) \log_2 \frac{Q(i)}{S(i)} \\ &= \Lambda \log_2 \frac{\Lambda}{\Lambda'} + (1-\Lambda) \log_2 \frac{1-\Lambda}{1-\Lambda'} + D(Q||S). \end{aligned} \tag{F.9}$$

The choice of $S(i)$ to minimize this expression is $S(i) = Q(i)$ for all i. This sets $D(Q\|S)$ at its minimum value of 0. Translating back from S to the original distribution R, we have $R(i) = (\Lambda'/\Lambda)Q(i)$ for $i \in U$; and $R(i) = ((1 - \Lambda')/(1 - \Lambda))Q(i)$ for $i \notin U$. This completes the first step in the proof.

The second step concerns choosing the optimal choice of Λ'. By definition, $\Lambda' = \sum_{i:R(i)f \leqslant Q(i)} R(i)$. Moreover, we know that Λ' is bounded by

$$0 \leqslant \Lambda' = \sum_{i:R(i)f \leqslant Q(i)} R(i) \leqslant \frac{1}{f} \sum_{i:R(i)f \leqslant Q(i)} Q(i) = \frac{\Lambda}{f}. \tag{F.10}$$

Let us view the quantity to be minimized as a function of Λ':

$$F(\Lambda') = \Lambda \log_2 \frac{\Lambda}{\Lambda'} + (1-\Lambda) \log_2 \frac{1-\Lambda}{1-\Lambda'}. \tag{F.11}$$

Differentiating and simplifying, we obtain

$$\frac{dF}{d\Lambda'} = \log_2 e \frac{dF}{d\Lambda'}\left(\Lambda \log_e \frac{\Lambda}{\Lambda'} + (1-\Lambda)\log_e \frac{1-\Lambda}{1-\Lambda'}\right),$$
$$\frac{dF}{d\Lambda'} = \log_2 e\left(\Lambda\left(-\frac{1}{\Lambda}\right) - (1-\Lambda)\left(-\frac{1}{1-\Lambda'}\right)\right),$$
$$\frac{dF}{d\Lambda'} = \log_2 e\left(\frac{\Lambda'-\Lambda}{(1-\Lambda')\Lambda'}\right). \quad \text{(F.12)}$$

By Eq. (F.10), we know that $\Lambda' < \Lambda$, which implies that $dF/d\Lambda' < 0$. This means that to minimize F, we should maximize Λ', which means that it should be set to its maximum value (again by Eq. (F.10)) $\Lambda' = \Lambda/f$.

Putting the results of steps 1 and 2 together, we have the result that the distribution R should be chosen as follows, in order to minimize the Kullback–Liebler distance with Q: $R(i) = ((\Lambda/f)/\Lambda)Q(i) = (Q(i)/f)$ for $i \in U$; and $R(i) = ((1-\Lambda/f)/(1-\Lambda))\,Q(i)$ for $i \notin U$, with the resulting minimum Kullback–Liebler distance:

$$D(P||Q) = \Lambda \log_2 f + (1-\Lambda)\log_2 \frac{1-\Lambda}{1-\Lambda/f} \quad \text{(F.13)}$$

We can bound this quantity as follows. We first note that this expression increases monotonically as f tends to infinity, which implies that

$$D(P||Q) \geqslant \Lambda \log_2 f + (1-\Lambda)\log_2(1-\Lambda). \quad \text{(F.14)}$$

A Taylor expansion of the right hand side of (F.14) gives

$$(1-\Lambda)\log_2(1-\Lambda)$$
$$= (1-\Lambda)\left(-\Lambda - \frac{\Lambda^2}{2} - \cdots - \frac{\Lambda^m}{m}\right)\log_2 e$$
$$\times\left(-\Lambda + \frac{\Lambda^2}{2.1} + \frac{\Lambda^3}{3.2} + \cdots + \frac{\Lambda^m}{m(m-1)}\right)$$
$$\times \log_2 e \geqslant -\Lambda \log_2 e. \quad \text{(F15)}$$

Putting these results together, we have

$$D(P||Q) \geqslant \Lambda \log_2 f - \Lambda \log_2 e = \Lambda \log_2 \frac{f}{e}. \quad \text{(F.16)}$$

This completes the proof of the Lemma. □

In proving the Lemma, we have considered arbitrary P and Q. We now consider the case where a sequence of $j-1$ words have made up the linguistic input so far, which we denote, x; and the distributions are the 'true' distribution $P_\mu(\cdot|x)$ (corresponding to P in (F.16)) and the learner's distribution $P_\lambda(\cdot|x)$ (corresponding to Q in (F.16)). Let us write the probability of an undergeneralization error after the sequence x as $\Lambda_i(x)$. The expected number undergeneralization errors at the jth word in the sequence, which we shall write $\langle \Lambda_j \rangle$, is the sum of the probabilities of such an error after x (i.e., $\Lambda_i(x)$) weighted by the probability of the initial sequence, x, (i.e., $P_\mu(x)$). Thus,

$$\langle \Lambda_j \rangle = \sum_{x:l(x)=j-1} P_\mu(x)\Lambda_j(x). \quad \text{(F.17)}$$

Applying the equation for the expected amount of information wasted in encoding the jth item, D_j (Eq. (20), in the main text):

$$D_j(P_\mu||P_\lambda) = \sum_{x:l(x)=j-1} P_\mu(x) D(P_\mu(\cdot|x)||P_\lambda(\cdot|x))$$
$$\geqslant \log_2 \frac{f}{e} \sum_{x:l(x)=j-1} P_\mu(x)\Lambda_j(x)$$
$$= \log_2 \frac{f}{e}\langle \Lambda_j \rangle, \quad \text{(F.18)}$$

where the inequality follows from (F.16) and the equality from (F.17).

If $f > e$, then $\log_2(f/e) > 0$, and hence we can divide through by this factor to give

$$\langle \Lambda_j \rangle \leqslant D_j(\mu||\lambda)\frac{1}{\log_2 f/e}. \quad \text{(F.19)}$$

Thus, the expected number of 'soft' undergeneralization errors for an infinite input sequence, where the probability of a sequence is underestimated by a factor $f > e$ is

$$\sum_{j=1}^{\infty}\langle \Lambda_j \rangle \leqslant \sum_{j=1}^{\infty} D_j(\mu||\lambda)\frac{1}{\log_2 f/e} \leqslant K(\mu)\frac{1}{\log_2 f/e}, \quad \text{(F.20)}$$

where the final inequality follows from Step 2 of the proof of the Prediction Theorem in the main text. This completes the proof. □

References

Akhtar, N., Callanan, M., Pullum, G., & Scholz, B. (2004). Learning antecedents for anaphoric one. *Cognition*, *93*, 141–145.

Angluin, D. (1980). Inductive inference of formal languages from positive data. *Information and Control*, *45*, 117–135.

Baker, C. L. (1979). Syntactic theory and the projection problem. *Linguistic Inquiry*, *10*, 533–581.

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

Blum, L., & Blum, M. (1975). Toward a mathematical theory of inductive inference. *Information and Control*, *28*, 125–155.

Bowerman, M. (1988). The 'no negative evidence' problem: How do children avoid constructing an overly general grammar? In J. A. Hawkins (Ed.), *Explaining language universals* (pp. 443–466). Oxford: Basil Blackwell.

Brent, M. R., & Cartwright, T. A. (1996). Distributional regularity and phonotactic7y constraints are useful for segmentation. *Cognition*, *61*, 93–126.

Brown, R., & Hanlon, C. (1970). Derivational complexity and order of acquisition in child speech. In J. Hayes (Ed.), *Cognition and the developmental of language* (pp. 11–53). New York: Wiley.

Chaitin, G. J. (1969). On the simplicity and speed of programs for computing infinite sets of natural numbers. *Journal of the Association for Computing Machinery*, *16*, 407–422.

Chaitin, G. J. (1987). Beyond Godel's proof. *IBM Research Magazine*, *25*, 12–15.

Chater, N. (1996). Reconciling simplicity and likelihood principles in perceptual organization. *Psychological Review*, *103*, 566–581.

Chater, N. (1997). Simplicity and the mind. *The Psychologist*, 495–498.

Chater, N. (1999). The search for simplicity: A fundamental cognitive principle? *Quarterly Journal of Experimental Psychology*, *52A*, 273–302.

Chater, N., & Vitányi, P. (2002). Simplicity: A unifying principle in cognitive science? *Trends in Cognitive Sciences*, *7*, 19–22.

Chater, N., & Vitányi, P. (2003). The generalized universal law of generalization. *Journal of Mathematical Psychology*, *47*, 346–369.

Chomsky, N. (1957). *Syntactic structures*. The Hague: Mouton.

Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

Chomsky, N. (1980). *Rules and representations*. Cambridge, MA: MIT Press.

Chomsky, N. (1981). *Lectures on government and binding*. Foris: Dordrecht.

Chomsky, N. (1986). *Barriers*. Cambridge, MA: MIT Press.

Chomsky, N. (1995). *The minimalist program*. Cambridge, MA: MIT Press.

Christiansen, M. H., & Chater, N. (1994). Generalization and connectionist language learning. *Mind and Language*, *9*, 273–287.

Christiansen, M. H., & Chater, N. (1999). Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science*, *23*, 157–205.

Crain, S., & Lillo-Martin, D. (1999). *Linguistic theory and language acquisition*. Oxford: Blackwell.

Dowman, M. (2000). Addressing the learnability of verb subcategorizations with Bayesian inference. In L. R. Gleitman, & A. K. Joshi (Eds.), *Proceedings of the 22nd annual conference of the Cognitive Science Society*. Mahwah, NJ: Erlbaum.

Ellison, M. (1992). The machine learning of phonological structure. Ph.D. thesis, University of Western Australia.

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

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*, *48*, 71–99.

Feldman, J. (1972). Some decidability results on grammatical inference and complexity. *Information and Control*, *20*, 244–262.

Fodor, J. D., & Crain, S. (1987). Simplicity and generality of rules in language acquisition. In B. MacWhinney (Ed.), *Mechanisms of language acquisition* (pp. 35–63). Hillsdale, NJ: Erlbaum.

Fodor, J. D., & Crowther, C. (2002). Understanding stimulus poverty arguments. *The Linguistic Review*, *19*, 105–145.

Gazdar, G., Klein, E., Pullum, G., & Sag, I. (1985). *Generalized phrase structure grammar*. Oxford: Blackwell.

Gibson, E., & Wexler, K. (1994). Triggers. *Linguistic Inquiry*, *25*, 407–454.

Gold, E. M. (1967). Language identification in the limit. *Information and Control*, *16*, 447–474.

Goldsmith, J. (2001). Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, *27*, 153–198.

Hahn, U., & Chater, N. (1998). Similarity and rules: Distinct? exhaustive? empirically distinguishable? *Cognition*, *65*, 197–230.

Harman, G. (1965). The inference to the best explanation. *Philosophical Review*, *74*, 88–95.

Harris, Z. (1951). *Methods in structural linguistics*. Chicago: Chicago University Press.

Hoekstra, T., & Kooij, J. G. (1988). The innateness hypothesis. In J. A. Hawkins (Ed.), *Explaining language universals*. Oxford: Basis Blackwell.

Horning, J. J. (1969). A study of grammatical inference. Technical Report CS 139, Computer Science Department, Stanford University.

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

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

Joshi, A., & Schabes, Y. (1997). Tree adjoining grammars. In G. Rozenberg, & A. Salomaa (Eds.), *Handbook of Formal Languages*, Vol. 3 (pp. 69–124). Berlin: Springer.

Koffka, K. (1962). *Principles of Gestalt psychology* (5th ed.). London: Routledge and Kegan Paul (Original work published in 1935).

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

Legate, J., & Yang, C. (2002). Empirical re-assessment of stimulus poverty arguments. *The Linguistic Review*, *19*, 151–162.

Li, M., & Vitányi, P. (1997). *An introduction to Kolmogorov complexity theory and its applications* (2nd ed.). Berlin: Springer.

Lidz, J., Waxman, S., & Freedman, J. (2003). What infants know about syntax but couldn't have learned: Evidence for syntactic structure at 18-months. *Cognition*, *89*, B65–B73.

Mach, E. (1959). *The analysis of sensations and the relation of the physical to the psychical*. New York: Dover Publications (Original work published 1886).

MacWhinney, B. (1993). The (il)logical problem of language acquisition. In *Proceedings of the 15th annual conference of the Cognitive Science Society* (pp. 61–70). Mahwah, NJ: Erlbaum.

MacWhinney, B. (2004). A multiple process solution to the logical problem of language acquisition. *Journal of Child Language*, *31*, 883–914.

Martin, E., & Osherson, D. N. (1998). *Elements of scientific inquiry*. Cambridge, MA: MIT Press.

van der Mude, A., & Walker, A. (1978). On the inference of stochastic regular grammars. *Information and Control*, *38*, 310–329.

Niyogi, P. (2006). The computational nature of language learning and evolution.

Onnis, L., Roberts, M., & Chater, N. (2002). Simplicity: A cure for overregularization in language acquisition. In L. R. Gleitman, & A. K. Joshi (Eds.), *Proceedings of the 24th Annual Conference of the Cognitive Science Society*. Mahwah, NJ: Erlbaum.

Osherson, D. N., Stob, M., & Weinstein, S. (1982). Ideal learning machines. *Cognitive Science*, *6*, 277–290.

Osherson, D. N., Stob, M., & Weinstein, S. (1984). Learning theory and natural language. *Cognition*, *17*, 1–28.

Osherson, D. N., Stob, M., & Weinstein, S. (1985). *Systems that learn*. Cambridge, MA: MIT Press.

Osherson, D. N., & Weinstein, S. (1982). A note on formal learning theory. *Cognition*, *11*, 77–88.

Paul, W. J., Seiferas, J. I., & Simon, J. (1981). An information theoretic approach to time bounds for on-line computation. *Journal of Computer and System Sciences*, *23*, 108–126.

Perfors, A., Tenenbaum, J. B., & Regier, T. (2006). Poverty of the stimulus? A rational approach. In: *Proceedings of the 28th Annual Conference of the Cognitive Science Society*. Mahwah, NJ, Erlbaum.

Pinker, S. (1979). Formal models of language learning. *Cognition*, *7*, 217–283.

Pinker, S. (1984). *Language learnability and language development*. Cambridge, MA: MIT Press.

Pitt, L. (1989). Probabilistic inductive inference. *Journal of the ACM*, *36*, 383–433.

Pullum, G., & Scholz, B. (2002). Empirical assessment of stimulus poverty arguments. *The Linguistic Review*, *19*, 9–50.

Quinlan, J. R., & Rivest, R. (1989). Inferring decision trees using the minimum description length principle. *Information and Computation*, *80*, 227–248.

Regier, T., & Gahl, S. (2004). Learning the unlearnable: The role of missing evidence. *Cognition*, *93*, 147–155.

Rissanen, J. (1987). Stochastic complexity. *Journal of the Royal Statistical Society, Series B*, *49*, 223–239.

Rissanen, J. (1989). *Stochastic complexity and statistical inquiry*. Singapore: World Scientific.

Shannon, C. E. (1951). Prediction and entropy of printed English. *Bell System Technical Journal*, *30*, 50–64.

Rohde, D. L. T., & Plaut, D. C. (1999). Language acquisition in the absence of explicit negative evidence: How important is starting small? *Cognition*, *72*, 68–109.

Solomonoff, R. J. (1964). A formal theory of inductive inference, Parts 1 and 2. *Information and Control*, *7*(1-22), 224–254.

Solomonoff, R. J. (1978). Complexity-based induction systems: Comparisons and convergence theorems. *IEEE Transactions on Information Theory*, *24*, 422–432.

Steedman, M. (1996). *Surface structure and interpretation*. Cambridge, MA: MIT Press.

Tomasello, M. (2004). Syntax or semantics? Response to Lidz et al. *Cognition*, *93*, 139–140.

Valiant, L. (1984). A theory of the learnable. *Journal of the ACM*, *27*, 1134–1142.

Vapnik, V. (1998). *Statistical learning theory*. New York, NY: Wiley.

Vitányi, P. M. B., & Li, M. (2000). Minimum description length induction, Bayesianism, and Kolmogorov complexity. *IEEE Transactions on Information Theory*, *IT-46*, 446–464.

Wallace, C. S., & Freeman, P. R. (1987). Estimation and inference by compact coding. *Journal of the Royal Statistical Society, Series B*, *49*, 240–251.

Wexler, K., & Culicover, P. (1980). *Formal principles of language acquisition*. Cambridge, MA: MIT Press.

Wolff, J. G. (1988). Learning syntax and meanings through optimisation and distributional analysis. In Y. Levy, I. M. Schlesinger, & M. D. S. Braine (Eds.), *Categories and processes in language acquisition* (pp. 179–215). Hillsdale, NJ: LEA.

Zurek, W. H. (Ed.). (1991). *Complexity, entropy, and the physics of information*. Redwood City, CA: Addison-Wesley.

Zvonkin, A. K., & Levin, L. A. (1970). The complexity of finite objects and the development of the concepts of information and randomness by mean of the theory of algorithms. *Russian Mathematical Surveys*, *25*, 83–124.