Identifying Empirical Laws

Martim Maria Mathias Cortez de Lobão
cortezdelobao@tecnico.ulisboa.pt

Técnico Lisboa, Lisboa, Portugal

November 2016

Abstract

Learning theory allows the study of the theoretical limits of the inductive inference of computable sets and functions. It is based on functions called scientists, which receive as input a sequence of observations (either points of the graph of a function or elements in a set) and return conjectures about the object to which those observations belong. A scientist is successful in identifying the object if there is a point in which it stabilizes on a correct conjecture. In this work, we cover a brief introduction to learning theory, including different types of learning environments (text, fat text, imperfect text, and informant) and different restrictions on the identificational power of scientists (noncomputable, computable, and memory-limited scientists). We establish the relation between the identification capacity of several kinds of scientist and learning environment, and we give examples of classes of sets and functions in each category. New methods to aid in the verification of whether or not a given class was identifiable were also created: Markov scientists for memory-limited identification and limit sets for noncomputable identification, which was developed based on the work of and in partnership with Professor José Félix Costa. Finally, we propose a concept for empirical identification so that the previously obtained results may be applicable in practice. This approach is inspired by the computable physical models defined in Szudzik [6]. We also create the concept of discoverable function using primitive recursive functions.
Keywords: scientist, set and function identification, identification in the limit, inductive inference, recursive function, discoverable function.

Introduction and philosophical framework

The initial step of the scientific method — the inductive step — hinges on the identification of patterns within observable facts. Pattern recognition is the necessary basis without which there could be no formulation of a scientific theory. It was the cyclical nature of the orbits of planets that eventually led to Kepler's laws, and it was Mendel's observation of hereditary traits in pea plants that prompted the foundation of genetics. The realization of the existence of patterns over randomness and order over chaos is what allows scientists to make accurate predictions of our world.

This process is akin to a game in which a player is required to guess the next number in a sequence by being told the first few elements. Upon learning that the first elements of a sequence are 2, 4, 6, 8, and 10, the player may conjecture that the following number will be 12, which is the smallest positive even number not yet in the sequence. If his guess turns out to be correct, he will likely maintain his hypothesis for the rule which governs the sequence, and if not, he will adjust his rule accordingly.

The crux of learning theory lies in understanding if and when it is possible to correctly induce general 'patterns' (e.g. an infinite sequence) using only its particular instantiations (e.g. a finite segment of the sequence). Just as with empirical science, it is in general not evident that it should even be possible to infer an infinite amount of information using only a finite amount of data. However, we will show that in many situations this limitation is not as severe as one would first assume. There are in fact several reasonable restrictions we may impose on the nature of the data which guarantee that many patterns (including a wide variety of functions and sets) are identifiable in an unbounded but finite amount of time.

1 Introduction to inductive inference

1.1 Preliminaries on computability

The reader should recall some fundamental computability definitions. We denote the classes of functions of type $\mathbb{N} \to \mathbb{N}$ of the partial recursive functions by $\boldsymbol{\mathcal{P}}$, of the recursive functions by $\boldsymbol{\mathcal{R}}$, of the primitive recursive functions by $\boldsymbol{\mathcal{PR}}$, and of the class of recursively enumerable sets by $\boldsymbol{\mathcal{E}}$.

Encodings and encoded functions

So far, we have only considered functions and sets within the natural numbers. To address this, we formulate a general procedure to 'encode' a countable set of numbers (e.g. $\mathbb{Q}$) into the set $\mathbb{N}$.

1.1 Definition (Encoding and representation)**.** We say that a function $f : S \to \mathbb{N}$ is an *encoding* of S if it is a bijection from S to $\mathbb{N}$. The inverse function $f^{-1} : \mathbb{N} \to S$ is called a *decoding* of S. If f is an encoding of S and $f(x) = n$, we say that n *represents* (or is the *representation* of) x.

Using encodings of various countable sets, we are able to use a partial recursive function defined in the natural numbers to encode any other 'computable' function whose domain and codomain are countable. *Encoded functions* are the extension of the concept of encodings from sets into functions. Similarly to encodings for sets, encoded functions allow the interpretation within the natural numbers of functions with a countable domain and codomain.

1.2 Definition (Encoded function)**.** Let A and B be countable sets, and let f and g be encodings of A and B respectively. If $\psi : A \to B$ is a total function and $\widetilde{\psi} : \mathbb{N} \to \mathbb{N}$ is a partial recursive function such that, for all $x \in A$, $\widetilde{\psi}(f(x)) = g(\psi(x))$, we say that $\widetilde{\psi}$ is the *encoded function* of ψ. In this case, we may frequently say that f is a (partial) recursive function when we mean that it has a (partial) recursive encoded function $\widetilde{\psi}$. Conversely, ψ is the *decoded function* of $\widetilde{\psi}$.

Indexes for sets and functions

Consider an encoding of A^*, where A is some finite alphabet. If we allow the set $\{a_0, a_1, \ldots, a_{k-1}\}$ to be the alphabet of symbols used in a specific programming language, we may use this encoding to associate a unique integer to each string of programming commands. In this way, it is possible to computably list out the class of computable functions $\boldsymbol{\mathcal{P}}$ using a *programming system.*

1.3 Definition (Programming system, Jain et al. [4])**.** A *programming system* for the class of partial recursive functions $\boldsymbol{\mathcal{P}}$ is a partial recursive function ν such that $\{\lambda x \,.\, \nu(\langle p, x\rangle) : p \in \mathbb{N}\} = \boldsymbol{\mathcal{P}}$. We may denote $\nu(\langle p, x\rangle)$ by $\nu_p(x)$.

A programming system entails the existence of a *universal* partial recursive function ν that encodes all other partial recursive functions. Such a function computes $\phi(x)$ for all $\phi \in \boldsymbol{\mathcal{P}}$ and $x \in \mathbb{N}$.

We say that a programming system is acceptable when there is an effective (recursive) function t that translates between it and any other acceptable programming system. In effect, this means that all acceptable programming systems are equivalent, in the sense that any partial recursive function that is expressible in one acceptable programming system is also expressible in another, and there is a recursive way of switching between both.

1.4 Definition (Index for a function and set)**.** With a fixed acceptable programming system ϕ, a ϕ-index p of a partial recursive function ϕ_p is simply called an *index* for ϕ_p. The domain of ϕ_p is the recursively enumerable set denoted by W_p, and p is an *index* for it.

Using indexes for (partial recursive) functions and (recursively enumerable) sets, it is possible to 'list out' all the elements of $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{E}}$. We call this list an *indexing.*

1.5 Definition (Indexing)**.** Using a fixed acceptable programming system ϕ, the acceptable indexing, or simply *indexing*, of all the partial recursive functions and the *indexing* of all the recursively enumerable sets are the lists $\{\phi_i\}_{i\in\mathbb{N}} = \phi_0, \phi_1, \phi_2, \ldots$ and $\{W_i\}_{i\in\mathbb{N}} = W_0, W_1, W_2, \ldots$, respectively.

It is important to note that indexings are not injective functions of $\mathbb{N}$ into $\boldsymbol{\mathcal{P}}$ or $\boldsymbol{\mathcal{E}}$. Whereas it is true that an integer is an index for a single function or set, a function or set can have more than one index. In fact, for any partial recursive function ψ or recursively enumerable set S, there are infinitely many indexes for ψ and S.

1.6 Lemma (Osherson et al. [5])**.** *Let $\phi_0, \phi_1, \ldots$ be an acceptable indexing of $\boldsymbol{\mathcal{P}}$. Then, for all $i \in \mathbb{N}$, the set $\mathrm{EQ}_i = \{j \in \mathbb{N} : \phi_j = \phi_i\}$ is infinite.*

1.2 Identification

Recall the guessing game given previously. Here, the reader played the role of an empirical scientist attempting to identify some natural law from within an infinite number of possible laws. This is the basis for the concept of identification: to conjecture hypotheses using a finite sequence of information. For each hypothesis (e.g. $\lambda x \,.\, x(x-1)$), we may assign to it an integer label (for example, using an appropriate encoding of the alphabet used).

1.7 Definition (Hypothesis)**.** A *hypothesis* is a natural number i.

A hypothesis may be thought of as the 'output' given by a scientist at any given time. The number i is simply the index of some partial recursive function ϕ_i. In order for a scientist to produce a hypothesis, it must be given some kind of 'input.'

Texts and prefixes

Texts are infinite sequences of natural numbers that respect certain properties. A text is a particular type of *learning environment*, which serve as the basis for the 'input' scientists receive.

1.8 Definition (Learning environment)**.** A *learning environment* is a mapping of $\mathbb{N}$ into $\mathbb{N} \cup \{\#\}$, where $\#$ is the blank symbol.

The *content* of any finite or infinite sequence — including texts and other learning environments — is simply the set of all natural numbers it contains.

1.9 Definition (Content)**.** Let s be a (possibly infinite) sequence of elements. The *content* of s, written *content* (s), is the set of all natural numbers that are elements of s.

Note that by the definition of content, $\#$ is never contained in the content of any learning environment, since it is not a natural number.

An important kind of learning environment is called a *text.* In order to refer to the content of a text for function, we introduce the notion of a *set representation.*

1.10 Definition (Set representation)**.** $S \subseteq \mathbb{N}$ is said to be the *set representation* of the set $\{(x, y) : \langle x, y \rangle \in S\}$ of ordered pairs, where $\langle x, y \rangle = n$ is the image of $(x, y,)$ through the Cantor pairing function. The set representation of the graph of a function f (or simply the set representation of f) is the set $\Psi(f) = \{\langle x, f(x) \rangle : x \in domain(f)\}$.

Texts are the simplest kind of learning environment, and they are defined *for* sets or functions.

1.11 Definition (Text, Gold [3])**.** A *text* T is a learning environment such that *(a)* T is a text *for* a recursively enumerable set S just in case *content* $(T) = S$, and *(b)* T is a text *for* a recursive function f just in case *content* $(T) = \Psi(f)$. The set of all texts (for both sets and functions) is denoted by text.

A text for a set S or a function f is then any sequence in which every element of S or every element of $\Psi(f)$ appears at least once, respectively. Note that texts for functions are only defined for recursive functions and not for partial recursive functions. Consequently, texts for functions are only defined for total functions.

1.12 Definition (Notation on texts)**.** We denote the $(n+1)$-th element of a text T by $T(n)$ and the initial sequence of length n of a text T is denoted by $T[n]$, with $n \geqslant 0$. Note that $T[n]$ does not include $T(n)$.

Despite the number of observations an empirical scientist can make being unlimited, it will always be limited to a finite number of observations at any given time. Similarly, a scientist learner can never 'work with' an entire text but only with a finite initial segment of a text, called a prefix.

1.13 Definition (Prefix)**.** A *prefix* σ is a finite initial segment $T[n]$ for some $n \in \mathbb{N}$ and some arbitrary text T, and we write $\sigma \subset T$. The set of all prefixes is $\textsc{pref} = \{T[n] : T \text{ is a text and } n \in \mathbb{N}\}$. The set of all prefixes for sets is denoted by $\textsc{seq}$ and the set of all prefixes for functions is denoted by $\textsc{seg}$.

1.14 Definition (Notation on prefixes)**.** Let $\sigma, \tau \in \textsc{pref}$ be two prefixes. Then

(a) the *length* of σ is the number of elements of σ, including blanks and repetitions. It is denoted by $|\sigma|$.

(b) the $(n+1)$-th element of σ is denoted by σ_n, for $n \geqslant 0$. Hence, $\sigma_n = T(n)$ when σ is a prefix of T and $n < |\sigma|$. (Note that the *first* element of σ is denoted σ_0, the second σ_1, and so on.)

(c) the sequence of the first n elements of σ is denoted by $\sigma[n]$, and the sequence of last n elements of σ is denoted by $\sigma[-n]$. If $|\sigma| = n$ then $\sigma[n] = \sigma[-n] = \sigma$.

(d) for $|\sigma| > 0$, the last member of σ — that is, $\sigma[-1] = \sigma_{|\sigma|-1}$ — is denoted by σ_{last}.

(e) for $|\sigma| > 0$, all of σ with its last element removed — that is, the prefix obtained by removing σ_{last} from σ — is denoted by σ^-. If $|\sigma| = 1$ then $\sigma^- = \varepsilon$.

(f) the result of concatenating τ onto the end of σ is denoted by $\sigma \diamond \tau$. Similarly, the result of concatenating a single character $x \in \mathbb{N} \cup \{\#\}$ onto the end of σ is written as $\sigma \diamond x$.

(g) if σ is an initial segment of τ, then we write $\sigma \subseteq \tau$. If σ is a proper initial segment of τ, then we write $\sigma \subset \tau$.

Prefixes — like texts — are sequences of natural numbers. Depending on the given context, these numbers may simply be elements of a set or may represent ordered pairs in a function. In the former case, we say that a prefix belongs to $\textsc{seq}$, and in the latter case, that it belongs to $\textsc{seg}$.

Scientists

A scientist is a function that receives a prefix of a text for a recursively enumerable set or a recursive function and returns a hypothesis for the index of the set or function in question. We will allow scientists to be possibly partial functions, as we do not require a guess to be made for every prefix. Additionally, we will permit a scientist to be a possibly noncomputable function.

1.15 Definition (Scientist, Gold [3])**.** A scientist for sets $\mathbf{M} : \textsc{seq} \to \mathbb{N}$ or a scientist for functions $\mathbf{M} : \textsc{seg} \to \mathbb{N}$ is a possibly partial and possibly noncomputable function.

A group of scientists is said to be a *school* of scientists.

1.16 Definition (School)**.** A *school of scientists*, or simply *school*, is a set of scientists for sets or for function, typically respecting a certain condition. If a school is a set of scientists for sets, we say it is a *set school*, and if it is a school for functions, we say it is a *function school.* The school of all scientists for sets is denoted by $\mathbb{S}$ and the school of all scientists for functions is denoted by $\mathbb{S}_f$.

For a scientist to identify a text (and, subsequently, a set or class of sets), we must define a notion of convergence for it.

1.17 Definition (Convergence to an index)**.** A scientist $\mathbf{M}$ is said to *converge to an index* i on text T just in case, for all but finitely many $n \in \mathbb{N}$, $\mathbf{M}(T[n]) = i$. That is, it converges if, for all but finitely many prefixes σ of T, $\mathbf{M}(\sigma) = i$.

If a scientist converges to a correct hypothesis for a text, then it is said to identify the text.

1.18 Definition (Identification)**.**

(a) A scientist $\mathbf{M}$ *identifies* a text T for a set S or for a function f just in case $\mathbf{M}$ converges to an index i and i is an index for S or an index for f, respectively.

(b) A scientist $\mathbf{M}$ *identifies* a set S or a function f just in case $\mathbf{M}$ identifies every text T for S or for f, respectively.

(c) A scientist $\mathbf{M}$ *identifies* a class $\mathcal{C}$ of sets or functions just in case $\mathbf{M}$ identifies every set $S \in \mathcal{C}$ or every function $f \in \mathcal{C}$, respectively.

The previous definition implies that there are several distinct levels of identification. However, identification is only meaningful when it refers to an entire class of sets or functions. For example, a scientist that is a constant function which returns the integer i for any prefix of any text it receives will trivially identify the set whose index is i. The crux of identification may therefore be said to be the ability to *distinguish* between multiple objects.

Learning environments

In addition to texts, we will also consider three other types of *learning environments*: fat texts, imperfect texts, and informants.

1.19 Definition (Fat text)**.** A *fat text* T is a learning environment such that

(a) T is a fat text *for* a recursively enumerable set S just in case *content* $(T) = S$ and for all $x \in S$, $\{n : T(n) = x\}$ is infinite,

(b) T is a text *for* a recursive function f just in case *content* $(T) = \Psi(f)$ and for all $\langle x, y\rangle \in \Psi(f)$, $\{n : T(n) = \langle x, y\rangle\}$ is infinite.

The set of all fat texts is denoted by $\mathsf{fat\ text}$.

1.20 Definition (Imperfect text)**.** An *imperfect text* T is a learning environment such that

(a) T is an imperfect text *for* a recursively enumerable set S just in case *content* $(T) = U$, where U is a finite variant of S[1] (i.e. if *content* $(T) = S \cup D - D'$, where both D and D' are finite),

(b) T is an imperfect text *for* a recursive function f just in case *content* $(T) = \Psi(g)$, where g is a total recursive function that is a finite variant of f.[1]

We denote the set of all imperfect texts by $\mathsf{imp.\ text}$.

1.21 Definition (Informant)**.** An *informant* T is a learning environment such that

(a) T is an informant *for* a recursively enumerable set S just in case T is a text for the characteristic function χ_S of S, i.e. for all $x \in S$, there is $n \in \mathbb{N}$ such that $\langle x, 1\rangle = T(n)$ and, for all $x \notin S$, there is $n \in \mathbb{N}$ such that $\langle x, 0\rangle = T(n)$,

(b) T is an informant *for* a recursive function f just in case T is a text for the characteristic function $\chi_{\Psi(f)}$ of the set representation of f, i.e. for all $x, y \in \mathbb{N}$, there is $n \in \mathbb{N}$ such that $\langle\langle x, y\rangle, 1\rangle = T(n)$ if $\langle x, y\rangle \in f$ and $\langle\langle x, y\rangle, 0\rangle = T(n)$ otherwise.

The set of all informants is denoted by $\mathsf{informant}$. The set of all prefixes of informants for sets is denoted ISEQ and the set of all prefixes of informants for functions is denoted ISEG.

Collections of Classes

Whereas a single scientist may be able to identify a single class, a set of scientists that respect a certain condition is able to identify a large group of classes of sets or functions.

For example, the school of all scientists for sets contains every possible scientist for sets, including noncomputable scientists. A *collection* of classes for some school of scientists $\mathbb{S}'$ and some learning environment e is the set of all classes which are identifiable by $\mathbb{S}'$ in e.

1.22 Definition (Collection)**.** Let $\mathbb{S}'$ be a school of scientists for sets or for functions and e be a learning environment. The set of all classes $\mathcal{C}$ such that there is some scientist $\mathbf{M} \in \mathbb{S}'$ that identifies $\mathcal{C}$ in the learning environment e is called the *collection* of classes, or simply *collection*, for $\mathbb{S}'$ and e, and is denoted by $[\mathbb{S}', \mathsf{e}]$.

2 Identification of sets and functions

2.1 Main results

One of the primary focuses of this chapter is producing many examples of classes of sets and functions that are or are not identifiable by a given school of scientists. Proving that a class is identifiable typically only requires explicitly constructing a scientist and verifying that it does indeed identify the class in question, but showing that a class *is not* identifiable is usually harder. In this first section, we explore several important results of identification theory that will be very useful in later proofs.

[1] We say that a set A is a finite variant of B if both $(A - B)$ and $(B - A)$ are finite. Similarly, a function f is a finite variant of a function g if the set of points $\{x \in \mathbb{N} : f(x) \neq g(x)\}$ where f and g differ is finite.

2.1.1 Locking sequences

A *locking sequence* provides a necessary condition for identification in all types of learning environments for both sets and functions. It relies on a fairly evident observation: if a scientist identifies some set or function, it will necessarily do so after finitely many observations. If, for every text for the given set or function beginning with this prefix σ, the scientist always returns the same output after reading σ, then σ is said to 'lock' the scientist into the correct hypothesis.

2.1 Definition (Locking sequence, Blum and Blum [2])**.**

1. Let $S \in \boldsymbol{\mathcal{E}}$ be a recursively enumerable set, $\mathbf{M}$ be a scientist for sets and $\sigma \in \textsc{seq}$ be a prefix of a text for sets. We say that a nonempty prefix σ is a *locking sequence* for $\mathbf{M}$ on S just in case *(a)* $\textit{content}\,(\sigma) \subseteq S$, *(b)* $W_{\mathbf{M}(\sigma)} = S$, and *(c)* for all $\tau \in \textsc{seq}$, if $\textit{content}\,(\tau) \subseteq S$, then $\mathbf{M}(\sigma \diamond \tau) = \mathbf{M}(\sigma)$.

2. Let $f \in \boldsymbol{\mathcal{R}}$ be a recursive function, $\mathbf{M}$ be a scientist for functions and $\sigma \in \textsc{seg}$ be a prefix of a text for functions. We say that σ is a *locking sequence* for $\mathbf{M}$ on f just in case *(a)* $\textit{content}\,(\sigma) \subseteq \Psi(f)$, *(b)* $\phi_{\mathbf{M}(\sigma)} = f$, and *(c)* for all $\tau \in \textsc{seg}$, if $\textit{content}\,(\tau) \subseteq \Psi(f)$, then $\mathbf{M}(\sigma \diamond \tau) = \mathbf{M}(\sigma)$.

A locking sequence is always defined with respect to a scientist and a set or function, so that a locking sequence for one scientist is in general not a locking sequence for another. Note that this implies that there do not exist locking sequences for particular texts: a sequence must be locking for *all* texts for a given set or function. Additionally, there is no requirement for a locking sequence to be as small as possible — it need not be a 'minimal' locking sequence. Indeed, if a prefix $\sigma \in \textsc{pref}$ is a locking sequence for a set S or function f, then any larger prefix $\tau \in \textsc{pref}$ such that $\sigma \subset \tau$ and $\textit{content}\,(\tau) \subseteq S$ or $\textit{content}\,(\tau) \subseteq \Psi(f)$ is also a locking sequence for S or f, respectively.

2.2 Theorem (Blum and Blum [2])**.** *Let* $\mathbf{M}$ *be a scientist for sets or for functions that identifies the set* $S \in \boldsymbol{\mathcal{E}}$ *or the function* $f \in \boldsymbol{\mathcal{R}}$*, respectively. Then there exists a prefix* $\sigma \in \textsc{seq}$ *or* $\sigma \in \textsc{seg}$ *that is a locking sequence for* $\mathbf{M}$ *on* S *or on* f*, respectively.*

2.1.2 Angluin's Theorem

Locking sequences provide a necessary condition for identification in text by any kind of scientist. In set identification, if we consider the content of a locking sequence, we may expect to obtain some sort of locking *set*. The precise result was proved by Angluin [1] and actually constitutes a necessary *and sufficient* condition for set identification in text by a scientist. In effect, it provides the upper bound of classes of sets that can be identified by any sort of scientist working in text.

2.3 Definition (Angluin condition, Angluin [1])**.** A class of sets $\mathcal{C}$ is said to respect the *Angluin condition* if, for every set $L \in \mathcal{C}$, there exists a finite set $D_L \subseteq L$ (called an Angluin set for L) such that, for all $L' \in \mathcal{C}$ where $L' \neq L$, if $D_L \subseteq L'$, then $L' \not\subset L$.

2.4 Angluin's Theorem (Angluin [1])**.** *A class of sets* $\mathcal{C}$ *is identifiable by a scientist in text if and only if it respects the Angluin condition, that is, if and only if every set* $L \in \mathcal{C}$ *has an Angluin set.*

2.1.3 Limit sets

Angluin's Theorem, while providing a necessary and sufficient condition for set identifiability, is not always an easy result to apply in practice. In general, it may be unclear whether a given set is an Angluin set. Thus, an alternative condition for identifiability is desirable. A possible condition stems from the topological notion of a *limit set.*

2.5 Definition (Limit set)**.** Let $\{S_i\}_{i\in\mathbb{N}}$ be a sequence of sets. We say that S is the *limit set* of $\{S_i\}_{i\in\mathbb{N}}$ (or that $\{S_i\}_{i\in\mathbb{N}}$ *converges to* S), and we write $\{S_i\}_{i\in\mathbb{N}} \to S$, if *(a)* for all $i \in \mathbb{N}$, $S_i \subsetneq S$, and *(b)* for each finite set $D \subseteq S$, there is an $i \in \mathbb{N}$ such that $D \subseteq S_i$.

We can show that Definition 2.5 is a necessary and sufficient condition for identifiability.

2.6 Proposition. *Let* S *be the limit set of a sequence* $\{S_i\}_{i\in\mathbb{N}}$ *of sets. Then a class* $\mathcal{C}$ *is identifiable if and only if it does not contain* $\{S_i\}_{i\in\mathbb{N}}$ *and* S*.*

2.2 Identification by general scientist

There exist countless constraints and requirements one may conceivably place on scientists. A 'general' scientist — or simply a 'scientist' — is a scientist without any such restrictions, and therefore the collection of classes which are identified by the school of general scientists represents the upper limit of what may be identifiable in a given learning environment. As we shall see, even scientists on which no restrictions are applied may still have limited identification power. This limitation therefore stems exclusively from the nature of inductive inference.

2.2.1 In sets

The collection of all classes of sets that are identifiable in text by the school of all scientists for sets — that is, $[\mathbb{S}, \mathsf{text}]$ — contains several different classes. One example is $\textsc{fin}$, the class of all finite sets, as so $\textsc{fin} \in [\mathbb{S}, \mathsf{text}]$.

2.7 Proposition. $\textsc{fin} \in [\mathbb{S}, \mathsf{text}]$

2.8 Proposition (Gold [3])**.** $\textsc{fin} \cup \{\mathbb{N}\} \notin [\mathbb{S}, \mathsf{text}]$

Note that while FIN is identifiable and the class consisting of only the set $\mathbb{N}$ is also (trivially) identifiable, the union of both classes is *not* identifiable.

2.9 Theorem. *The collection* $[\mathbb{S}, \mathsf{text}]$ *is not closed under unions.*

Informants provide a much more robust learning environment for scientists, enabling the identification of a much larger class of sets. Indeed, the additional power imparted by informants is enough to enable the identification of the entire class of recursively enumerable sets by a general scientist.

2.10 Proposition. $\boldsymbol{\mathcal{E}} \in [\mathbb{S}, \mathsf{informant}]$

2.11 Corollary. $[\mathbb{S}, \mathsf{text}] \subset [\mathbb{S}, \mathsf{informant}]$

For imperfect text, a general scientist for sets can identify a strictly smaller collection of classes. The following lemma provides a necessary condition for identifiability in imperfect text, and allows us to given examples of a several classes that are not identifiable in imperfect text.

2.12 Lemma. *Let* $\mathcal{C}$ *be a class of sets such that* $L \in \mathcal{C}$ *and* $L' \in \mathcal{C}$ *are finite variants. Then* $\mathcal{C} \notin [\mathbb{S}, \mathsf{imp.\ text}]$.

2.13 Corollary. $[\mathbb{S}, \mathsf{imp.\ text}] \subset [\mathbb{S}, \mathsf{text}]$

2.2.2 In functions

Function identification by scientists in text is powerful enough to identify the entire set $\boldsymbol{\mathcal{R}}$ of recursive functions.

2.14 Theorem (Jain et al. [4, p. 52])**.** $\boldsymbol{\mathcal{R}} \in [\mathbb{S}_f, \mathsf{text}]$

As with sets, function identification in imperfect text also implies a loss of identifying power relative to identification in text.

2.15 Proposition. $[\mathbb{S}_f, \mathsf{imp.\ text}] \subset [\mathbb{S}_f, \mathsf{text}]$

2.3 Identification by computable scientist

Attempts made at developing artificial 'scientists' involve using computers or similar machines to automatically learn or identify scientific laws. Indeed, even machines such as quantum computers cannot perform any operation that may not also be performed by a Turing machine. Thus, if we wish to analyze these machines, it is natural to impose a computability restriction on our study of scientists.

2.16 Definition (Computable scientist)**.** A scientist is said to be *computable* if it can be simulated by a deterministic Turing machine. We denote the school of computable scientists for sets by $\mathbb{S}^{\text{CP}}$ and the school of computable scientists for functions by $\mathbb{S}_f^{\text{CP}}$.

2.3.1 In sets

Many sets that are identifiable by a general scientist are also identifiable by a computable scientist. This includes simply classes such as FIN, and so FIN $\in [\mathbb{S}^{\text{CP}}, \mathsf{text}]$. However, there do exist sets which are identifiable by a general scientist but are not computably identifiable. We present such a class of sets in Proposition 2.18.

2.17 Definition (Set K)**.** The set $\{i \in \mathbb{N} : i \in W_i\}$ is denoted K.

2.18 Proposition (Osherson et al. [5, p. 48])**.** *Let* $\mathcal{K}^* = \{K \cup \{x\} : x \in \mathbb{N}\}$. *Then* $\mathcal{K}^* \in [\mathbb{S}, \mathsf{text}]$ *but* $\mathcal{K}^* \notin [\mathbb{S}^{\text{CP}}, \mathsf{text}]$, *i.e.* $\mathcal{K}^*$ *is identifiable in text by a noncomputable scientist but not by a computable scientist.*

2.19 Proposition. $[\mathbb{S}^{\text{CP}}, \mathsf{text}] \subset [\mathbb{S}, \mathsf{text}]$

This computational barrier continues to persist even for computational scientists operating on informant.

2.20 Proposition (Osherson et al. [5, p. 48])**.** $\boldsymbol{\mathcal{E}} \notin [\mathbb{S}^{\text{CP}}, \mathsf{informant}]$

2.21 Corollary. $[\mathbb{S}^{\text{CP}}, \mathsf{informant}] \subset [\mathbb{S}, \mathsf{informant}]$

2.3.2 In functions

An example of two simple classes of functions which are identifiable by a computable scientist in text is SD and AEZ.

2.22 Proposition. *Let* SD $= \{\phi_i \in \boldsymbol{\mathcal{R}} : \phi_i(0) = i\}$ *be the class of functions that are* self-defining. *Then* SD $\in [\mathbb{S}_f^{\text{CP}}, \mathsf{text}]$.

2.23 Proposition. *Let* AEZ $= \{\phi_i \in \boldsymbol{\mathcal{R}} :$ *there exists a finite set* $D \subset \mathbb{N}$ *such that* $\phi_i(x) = 0$ *for all* $x \notin D\}$ *be the class of functions that are* almost everywhere zero. *Then* AEZ $\in [\mathbb{S}_f^{\text{CP}}, \mathsf{text}]$.

Although both SD and AEZ are identifiable in text by a computable scientist, their union is not.

2.24 Theorem (The Nonunion Theorem, Blum and Blum [2])**.** SD $\cup$ AEZ $\notin [\mathbb{S}_f^{\text{CP}}, \mathsf{text}]$.

2.25 Corollary. $\boldsymbol{\mathcal{R}} \notin [\mathbb{S}_f, \mathsf{text}]$

2.26 Corollary. $[\mathbb{S}_f^{\text{CP}}, \mathsf{text}] \subset [\mathbb{S}_f, \mathsf{text}]$

This result shows that enforcing a computability constraint on scientists for functions in text implies that a strictly smaller collection of classes is identifiable. Nonetheless, many complex functions are still identifiable by a computable scientist in text. An example of such a class is $\boldsymbol{\mathcal{PR}}$, the class of all primitive recursive functions with domain and range in $\mathbb{N}$.

2.27 Proposition. $\boldsymbol{\mathcal{PR}} \in [\mathbb{S}_f^{\text{CP}}, \mathsf{text}]$

2.4 Identification by memory-limited scientist

It seems clear that a scientist — whether it be of human or of mechanical nature — must have a limited memory for the list of observations or experiments with which it is presented. As soon as each observation has finished being taken into consideration and possibly triggered a change in hypothesis, it is forgotten and erased from the scientist's memory. In this section, we consider the possible loss in identifying power imposed by such a limitation.

2.28 Definition (Memory-limited scientist)**.** A scientist $\mathbf{M}$ is said to be *memory-limited* just in case for all nonempty prefixes $\sigma, \tau \in \textsc{pref}$, if $\mathbf{M}(\sigma^-) = \mathbf{M}(\tau^-)$ and $\sigma_{\text{last}} = \tau_{\text{last}}$, then $\mathbf{M}(\sigma) = \mathbf{M}(\tau)$. We denote the school of memory-limited scientists for sets by $\mathbb{S}^{\text{ML}}$ and the school of memory-limited scientists for functions by $\mathbb{S}^{\text{ML}}_f$.

The concept of a memory-limited scientist is akin to the one of memorylessness commonly used in statistics and probability theory. Intuitively, a memory-limited scientist is one whose future hypotheses depend only future observations and on the scientist's present state.

A useful way of specifying a scientist which guarantees that it is memory-limited is to use a *Markov scientist*.

2.29 Definition (Markov scientist and memory function)**.** Let $s : \mathbb{N}^2 \to \mathbb{N}$ be a recursive, one-to-one function such that, for all $i, m \in \mathbb{N}$, $\omega(i, m)$ is an index for W_i (or ϕ_i, depending on context).[2] Additionally, let $\omega_1 : \mathbb{N} \to \mathbb{N}$ and $\omega_2 : \mathbb{N} \to \mathbb{N}$ be the inverse projections of ω, such that $\omega_1(k) = i$ and $\omega_2(k) = m$ if and only if $\omega(i, m) = k$.

We say that a scientist $\mathbf{M}$ is a *Markov scientist* just in case there is a function $\mu : (\mathbb{N} \cup \{\#\}) \times \mathbb{N}^2 \to \mathbb{N}^2$ (called the *memory function* of $\mathbf{M}$) such that, for all $\sigma \in \textsc{pref}$, *(a)* $\mathbf{M}(\sigma) = \omega(\mu_0)$, where $\mu_0 = (i_0, m_0)$ and $i_0, m_0 \in \mathbb{N}$, if $\sigma = \varepsilon$, and *(b)* $\mathbf{M}(\sigma) = \omega(\mu(\sigma_{\text{last}}, i, m))$, where $i = \omega_1(\mathbf{M}(\sigma^-))$ and $m = \omega_2(\mathbf{M}(\sigma^-))$, if $\sigma \neq \varepsilon$.

The memory function μ of a Markov scientist $\mathbf{M}$ may be thought of as the process through which a memory-limited scientist obtains a new hypothesis. It takes three inputs — σ_{last}, i, and m — where i represents the conjecture for the set or function that $\mathbf{M}$ gives on σ^- and m acts as the 'memory' used by $\mathbf{M}$. Intuitively, a Markov scientist $\mathbf{M}$ only converges if both its conjecture i and its memory m also converge to an integer. Thus, $\mathbf{M}$ identifies a set L or a function f if and only if it converges to $\omega(i, m)$, where $W_i = L$ or $\phi_i = f$, respectively.

A necessary (but not sufficient) condition for the convergence of a Markov scientist is the existence of a 'fixed point' of its memory function.

[2]Such an ω must exist by the s-m-n Theorem.

2.30 Definition (Fixed point)**.** Let μ be the memory function of a Markov scientist. We say that (i, m) is a *fixed point* of μ just in case there is some $n \in \mathbb{N}$ such that $\mu(n, i, m) = (i, m)$.

2.31 Lemma. *If a Markov scientist* $\mathbf{M}$ *identifies some set or function, then its memory function* μ *has at least one fixed point.*

The following theorem shows that the definition of a Markov scientist is equivalent to the one of a memory-limited scientist.

2.32 Theorem. *Let* $\mathcal{C}$ *be a class of sets or functions. Then there is a memory-limited scientist* $\mathbf{M}$ *that identifies* $\mathcal{C}$ *in text if and only if there is a Markov scientist* $\mathbf{K}$ *that identifies* $\mathcal{C}$ *in text.*

2.4.1 In sets

The restriction of memory limitation on scientists for sets in text provokes a real loss of identifying power when compared to general scientists for sets in text.

2.33 Proposition. $[\mathbb{S}^{\text{ML}}, \mathsf{text}] \subset [\mathbb{S}, \mathsf{text}]$

The previous proposition clearly shows that memory limitation reduces the identifying power of scientists for sets in text. However, the following proposition states that this loss can be entirely compensated by working instead in fat text.

2.34 Proposition. $[\mathbb{S}^{\text{ML}}, \mathsf{fat\ text}] = [\mathbb{S}, \mathsf{text}]$

One might wonder what effect a composition of constraints will have on the identifying power of a scientist. For example, it may seem reasonable to assume that if a particular class is identifiable by both a computable scientist and by a memory-limited scientist, then there must also exist some computable, memory-limited scientist which identifies the same class. However, this is not the case, as is shown by the following proposition.

2.35 Proposition. *Consider the class* $\mathcal{L}_A$ *made up of the set* $L = \{\langle 0, i\rangle : i \in A\}$*, where* A *is a non-decidable, recursively enumerable set, the set* $L_n = \{\langle 1, n\rangle \cup L\}$ *and the set* $L'_n = \{\langle 1, n\rangle, \langle 0, n\rangle\} \cup L$*. Then (a)* $\mathcal{L}_A \in [\mathbb{S}^{\text{ML}}, \mathsf{text}]$*, (b)* $\mathcal{L}_A \in [\mathbb{S}^{\text{CP}}, \mathsf{text}]$*, and (c)* $\mathcal{L}_A \notin [\mathbb{S}^{\text{CP}} \cap \mathbb{S}^{\text{ML}}, \mathsf{text}]$.

2.36 Corollary. $[\mathbb{S}^{\text{CP}} \cap \mathbb{S}^{\text{ML}}, \mathsf{text}] \subset [\mathbb{S}^{\text{CP}}, \mathsf{text}] \cap [\mathbb{S}^{\text{ML}}, \mathsf{text}]$

2.4.2 In functions

As with sets, memory-limitation implies a strict loss of identifying power in functions.

2.37 Proposition. $[\mathbb{S}^{\text{ML}}_f, \mathsf{text}] \subset [\mathbb{S}_f, \mathsf{text}]$

3 Identifying scientific laws

3.1 The nature of empirical laws

Until now, we have only considered function and set identification under the premise that every element of a text was exact and did not include any measurement errors. In reality, the nature of the physical universe implies that experimental observations are not as well-defined as points on a function or elements of a set. Purely mathematical objects are dissimilar to the experimental data collected by real-world scientists, and so it is convenient to consider an identification paradigm which takes into account these kinds of errors.

We propose a new kind of empirical text and empirical scientist, inspired on the aforementioned concepts, and closely related to the *computable physical models* presented in Szudzik [6].

The largest obstacle we face in specifying a suitable model of empirical identification is a means of encoding the real numbers into the natural numbers. Such an objective is evidently impossible, as there is an uncountable infinity of real numbers but only a countable infinity of natural numbers. We therefore choose as our basis a subset of the computable numbers, the *primitive recursive numbers*, which are both countable and computably enumerable, and use them as the foundation of our model of empirical identification.

3.1 Definition (Primitive recursive number and primitive recursive defining function)**.** Let $x \in \mathbb{R}$. We say that x is a *primitive recursive number* if there exists a primitive recursive function $\gamma : \mathbb{N} \to \mathbb{N}$ such that

(a) $\gamma(0) = \max\{0, \operatorname{sgn}(x)\} = \begin{cases} 0, & \text{if } x \leqslant 0 \\ 1, & \text{if } x > 0, \end{cases}$

(b) $\gamma(n) = \lfloor 10^{n-1} \times |x| \rfloor$ for $n > 0$,

where sgn is the signum function. We say that γ is the *primitive recursive defining function* for x, and in general we denote γ by d_x. The set of all primitive recursive numbers is denoted by $\mathbb{P}$.

Thus, a real number x is said to be primitive recursive if there is some primitive recursive function γ such that $\gamma(0)$ encodes the sign of x, and, for all $n > 0$, $\gamma(n)$ encodes x up to $n-1$ decimal places.

3.2 Example. All algebraic numbers are primitive recursive. The number π is primitive recursive, and the first 5 terms of its primitive recursive defining are

$$d_\pi(0) = 1, \quad d_\pi(1) = 3, \quad d_\pi(2) = 31,$$
$$d_\pi(3) = 314, \quad d_\pi(4) = 3141.$$

We define an indexable set of *primitive real recursive functions* based on primitive recursive functions. These functions will form the basis of empirical identification.

3.3 Definition (Primitive real recursive function)**.** Let $\gamma : \mathbb{N}^2 \to \mathbb{N}$ be a primitive recursive function. We say that γ is a *primitive real recursive function* if, for each fixed $m \in \mathbb{N}$ and for all $n \in \mathbb{N}$, we have $\gamma(m, n) = d_y(n)$ for some primitive recursive number y. In other words, for each fixed $m \in \mathbb{N}$, $\lambda n \,.\, \gamma(m, n)$ is the primitive recursive defining function for some $y \in \mathbb{P}$.

Essentially, a primitive real recursive function will enable us to encode a range of primitive recursive defining functions into a single function of type $\mathbb{N}^2 \to \mathbb{N}$. Informally speaking, m will represent the 'independent variable' of the phenomenon that the scientist will attempt to identify, and $\lambda n \,.\, \gamma(m, n)$ will represent the decimal expansion of its 'dependent variable.'

3.4 Example. Table 1 shows values of the primitive real recursive function γ, where each line with a fixed m contains successive values of the primitive recursive defining function $d_y(n) = \lambda n \,.\, \gamma(m, n)$ for $y = m\pi$.

γ		n = 0	1	2	3	4
m	0	0	0	0	0	0
	1	1	3	31	314	3141
	2	1	6	62	628	6283
	3	1	9	94	942	9424
	4	1	12	125	1256	12566

Table 1: Values for a primitive real recursive function $\gamma(m, n)$.

We may use our indexation of the partial recursive functions to obtain an index for each primitive real recursive function.

3.5 Definition (Index for a primitive real recursive function)**.** Let $\gamma : \mathbb{N}^2 \to \mathbb{N}$ be a primitive real recursive function and $\phi : \mathbb{N} \to \mathbb{N}$ be the recursive function such that, for all $m, n \in \mathbb{N}$, $\gamma(m, n) = \phi_i(\langle m, n \rangle)$. If i is an index for ϕ then we say that i is also an index for γ.

3.6 Corollary. *There exists a recursive enumeration of the primitive real recursive functions.*

Using primitive real recursive functions, we may study the identification of functions of primitive recursive numbers. We call these *discoverable functions*, in the sense that one may 'discover' each one by resorting to its index. Discoverable functions are based on a particular type of function which we call *distinguishable function*.

3.7 Definition (Distinguishable function)**.** Let $f : \mathbb{R} \to \mathbb{R}$ be a total real function. We say that f is a *distinguishable function* just in case $f(\mathbb{Q}) \subseteq \mathbb{P}$, i.e. if, for all $x \in \mathbb{Q}$, $f(x)$ is a primitive recursive number.

3.8 Definition (Discoverable function)**.** Let $F : \mathbb{R} \to \mathbb{R}$ be a distinguishable function and $f : \mathbb{Q} \to \mathbb{P}$ be the restriction of F to $\mathbb{Q}$. We say that f is a *discoverable function* if there exists a primitive real recursive function $\gamma : \mathbb{N}^2 \to \mathbb{N}$ such that, for all $m, n \in \mathbb{N}$, $\gamma(m, n) = d_y(n)$, where $y = f(x) \in \mathbb{P}$ and $x = q^{-1}(m) \in \mathbb{Q}$. In this case, we say that γ is the primitive real recursive function of f. The class of all discoverable functions is denoted $\boldsymbol{\mathcal{DS}}$.

In other words, a function $f : \mathbb{Q} \to \mathbb{P}$ is discoverable if there is a primitive real recursive function γ such that $\lambda n \,.\, \gamma(m, n)$ is the primitive recursive defining function for each $f(x)$, with $x \in \mathbb{Q}$ and $q(x) = m$. Note that discoverable functions (which are of type $\mathbb{Q} \to \mathbb{P}$) are not a subset of distinguishable functions (which are of type $\mathbb{R} \to \mathbb{R}$), but rather that the latter is associated with the former on a many-to-one basis.

As before, we may use the indexation of the partial recursive functions to obtain an index for each discoverable function.

3.9 Definition (Index for a discoverable function)**.** Let f be a discoverable function and γ be its primitive real recursive function. If i is an index for γ then i is also an index for f.

3.10 Corollary. *There exists a recursive enumeration of the discoverable functions.*

The nature of empirical measurements

We must now consider what we wish to mean by 'empirical text.' In real-world experiments, scientists do not make exact measurements: there always exists some error or uncertainty associated to each one.

The nature of these errors suggests that empirical measurements are not point values but rather intervals. Indeed, the use of error bars is a common to graphically represent the amount of uncertainty in each measured value. In this way, we may think of each measurement as an interval of two rational numbers.

We assume that each dependent variable is always exact, and that the error in this assumption propagates into the error of the independent variable. Note that since the error in any measurement is always finite, there always exists some sufficiently large error margin such that the 'true' value lies within the margin. Of course, a scientist must always take care in selecting an suitably large error interval which is guaranteed to contain the real value.

One final assumption is that it is possible to make an arbitrarily precise measurement, and that given enough 'time' (where *time* can be taken to mean the number of experiments carried out by some scientist, technological advancements that enable the collection of more accurate data, or even literal time), such a measurement will eventually be made.

Using this model, we may define an *empirical text* for a total function f as a sequence of encoded triplets in $\mathbb{Q}^3$ that meet certain specific conditions.

3.11 Definition (Empirical text)**.** Let $f : \mathbb{R} \to \mathbb{R}$ be a total function and q be a fixed encoding of $\mathbb{Q}$. We say that a learning environment T is an *empirical text* for f if:

(a) for all elements $n \in T$, n is of the form $\langle q(x), \langle q(y_1), q(y_2)\rangle\rangle$, where $x, y_1, y_2 \in \mathbb{Q}$ and $y_1 \leqslant y_2$. We write $\langle x, y_1, y_2\rangle$ as shorthand for $\langle q(x), \langle q(y_1), q(y_2)\rangle\rangle$.

(b) if $\langle x, y_1, y_2\rangle \in T$ and $\langle x, y_1', y_2'\rangle \in T$ then $[y_1, y_2] \cap [y_1', y_2'] \neq \varnothing$.

(c) for all $x \in \mathbb{Q}$ and for all $\varepsilon > 0$, there is $\langle x, y_1, y_2\rangle \in T$ such that $f(x) - \varepsilon < y_1 \leqslant f(x) \leqslant y_2 < f(x) + \varepsilon$.

We denote the set of all empirical texts by `emp. text` and the set of all prefixes for empirical text by ESEG.

Elements of empirical texts can be thought of as the error bars shown in Figure 1. Here, we think of the x-variable as being an exact, rational measurement, and the pair (y_1, y_2) of elements from the measured variable y as encompassing the entirety of inaccuracies in each experiment or observation. In reality, since the x variable always has some level of error associated to it, this uncertainty is passed along to the measured variable with a suitable increase of each Δy. Put another way, an element $\langle x, y_1, y_2\rangle$ in an empirical text for a function f can simply be read as *it is absolutely certain that* $f(x)$ *is in the interval* $[y_1, y_2]$.

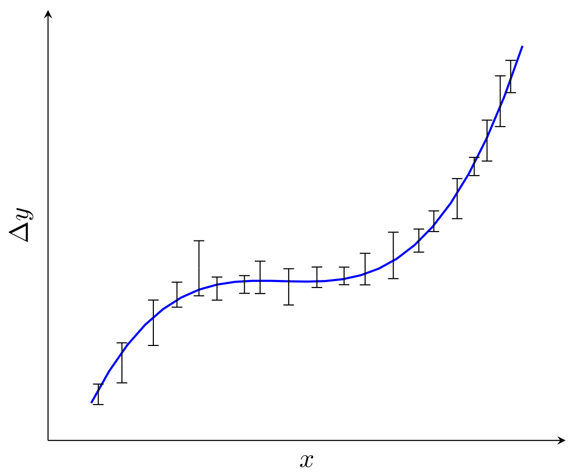

Figure 1: Measurements of an empirical text.

Empirical identification

With these concepts in mind, all that remains is to define an empirical scientist and to suitably adapt the previous definitions of identification to account for empirical identification.

3.12 Definition (Empirical scientist)**.** An *empirical scientist* $\mathbf{M} : \text{ESEG} \rightarrow \mathbb{N}$ for functions is a computable and possibly partial function. The school of empirical scientists is denoted $\mathbb{S}_e^{\text{CP}}$.

3.13 Definition (Empirical identification)**.** We say that an empirical scientist $\mathbf{M}$ *empirically identifies*, or simply *identifies*, a class of discoverable functions $\mathcal{C}$ if, for all $f \in \mathcal{C}$ and for all empirical texts T for f, $\mathbf{M}$ converges to an index for f. The collection of all classes of functions which are identifiable by a computable empirical scientist in empirical text is $[\mathbb{S}_e^{\text{CP}}, \mathsf{emp.\ text}]$.

3.2 Empirical identification of the class of discoverable functions

The most significant result of empirical identification is that the entirety of the class of discoverable functions is identifiable. These functions appear to at least constitute the majority of the scientific laws that have ever been discovered throughout history. They include Boyle's Law, Einstein's equations for General Relativity, Maxwell's equations, and many others. In fact, we are not aware of any functional scientific law which cannot be expressed or computed using discoverable functions.

3.14 Theorem. $\mathcal{DS} \in [\mathbb{S}_e^{\text{CP}}, \mathsf{emp.\ text}]$

4 Conclusion

The results presented in Chapter 2 present a concise overview of the classes of sets and functions which can be identified 'in the limit.' For a scientist for functions without any restrictions such as memory limitation or computability, it is possible to identify any recursive function in finite time. If we additionally require that scientists be computable, then the collection of classes of functions and of sets is strictly smaller.

However, these results are merely the theoretical limits or upper bounds to scientific identification. Indeed, aside from imperfect text, there are no considerations made about any kind of error (whether they be systematic or random) that are inherent to scientific observations. Chapter 3 therefore presents an adaptation of the concept of scientist in order to deal with errors of the sort usually encountered in experiments.

The applicability of the results of empirical identification presented in Chapter 3 is also heavily dependent on several assumptions which arguably do not reflect real-world situations. Empirical text in particular can be the object of many valid criticisms. For example, it may not be the case that errors in measurements necessarily converge to zero. It can also be argued that it is in general not possible to specify a discrete error bound for measurements, requiring that errors be considered as a random variable instead.

Certain topics may be worth further exploration in future works. For example, it is not clear how — or even if — we can extend the concept of empirical scientists to areas of science that conjecture theories which are not expressed mathematically, such as the theory of genetics or evolution. On the other hand, it is possible to continue improving the concept of empirical identification. One suggestion would be to construct a kind of empirical text that better reflects the randomness in error measurements, and another would be to devise a suitable definition for discoverable functions based on computable numbers instead of the primitive recursive numbers, thereby enriching the universe of functions that may be studied by empirical scientists.

Acknowledgements

I would like to thank my supervisor, Professor José Félix Costa, for his invaluable advice, unfailing availability, and infinite patience. This thesis would not have been possible without his guidance.

To Manel, for finding my motivation when I could not.

To my family, in particular to my parents and my grandmother, for their unconditional support.

And to Carolina, whose love and kindness have been my safe haven.

References

[1] Angluin, Dana. Inductive inference of formal languages from positive data. *Information and Control*, 45(2):117–135, 1980.

[2] Blum, Lenore and Blum, Manuel. Toward a mathematical theory of inductive inference. *Information and Control*, 28(2):125–155, 1975.

[3] Gold, E. Mark. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[4] Jain, Sanjay, Osherson, Daniel, Royer, James S., and Sharma, Arun. *Systems That Learn: An Introduction to Learning Theory.* MIT Press, Cambridge, Massachusetts, second edition, 1999.

[5] Osherson, Daniel N., Stob, Michael, and Weinstein, Scott. *Systems That Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists.* MIT Press, Cambridge, Massachusetts, 1986.

[6] Szudzik, Matthew P. The Computable Universe Hypothesis. *A Computable Universe*, pages 479–523, 2013. `arXiv:1003.5831v6`.