Inductive learning with corroboration

Phil Watson
Computing Laboratory
University of Kent at Canterbury
Canterbury
Kent CT2 7NZ
United Kingdom
P.R.Watson@ukc.ac.uk

May 24, 1999

Abstract

The basis of inductive learning is the process of generating and refuting hypotheses. Natural approaches to this form of learning assume that a data item that causes refutation of one hypothesis opens the way for the introduction of a new (for now unrefuted) hypothesis, and so such data items have attracted the most attention. Data items that do not cause refutation of the current hypothesis have until now been largely ignored in these processes, but in practical learning situations they play the key role of *corroborating* those hypotheses that they do not refute.

We formalise a version of K.R. Popper's concept of *degree of corroboration* for inductive inference and utilise it in an inductive learning procedure which has the natural behaviour of outputting the most strongly corroborated (non-refuted) hypothesis at each stage. We demonstrate its utility by providing characterisations of several of the commonest identification types. In many cases we believe that these characterisations make the relationships between these types clearer than the standard characterisations. The idea of learning with corroboration therefore provides a unifying approach for the field.

Keywords: Degree of Corroboration; Inductive Inference; Philosophy of Science.

1 Introduction

The field of machine inductive inference has developed in an ad hoc manner, in particular in the characterisations of identification types which have been achieved. In this paper we wish to propose a new unifying framework for the field based on the philosophical work of K. R. Popper, and in particular his concept of *degree of corroboration*. We will demonstrate that many of the existing identification types in the case of learning from text allow an alternative characterisation using the concept of learning with corroboration; in particular this approach reveals the existence of canonical learning algorithms for the various types.

In the next section we cover the basics of inductive learning. In Section 3 we cover as much of Popper's logic of scientific discovery as necessary for our purposes, and in Section 4.1 we treat his concept of degree of corroboration in more detail. In Section 5 we define the basics of an inductive learner with corroboration, and in Section 6 we give characterisations of many of the standard identification types using these learners. Section 7 contains some examples of the use of learning with corroboration in practice. Section 8 discusses some recent work of Gillies which has relevance, while Section 9 contains our conclusions and directions for further work.

2 Preliminaries

As usual $\mathbb{N}$ will denote the set of natural numbers, $\cup$ and $\cap$ will be set union and intersection respectively, while $\subseteq$ and $\subset$ will be the subset and proper subset relations respectively. We write $A \subseteq_{fin} B$ if A is a finite subset of B. The cardinality of the set A is written $\mid A \mid$ and the length of a sequence t is written $\mid t \mid$. Ambiguity will be resolved by context.

By Σ we denote any fixed finite alphabet of symbols. Let Σ^* be the free monoid over Σ, i.e. the set of all finite words (strings) produced using that alphabet. Any subset $L \subseteq \Sigma^*$ is called a language. We set $\overline{L} = \Sigma^* \setminus L$. Let L be a language and $t = s_0, s_1, s_2, \ldots$ an infinite sequence (possibly with repetitions) of strings from Σ^* such that $L = \{s_k \mid k \in \mathbb{N}\}$; then t is said to be a *text* for L (or, synonymously, a *positive presentation* of L) written $t \in Txt(L)$. If $\mathcal{L}$ is a class of languages and $(\exists L \in \mathcal{L})t \in Txt(L)$ then we write $t \in Txts(\mathcal{L})$. We refer to the initial segment of t of length $n+1$ by t_n, i.e. $t_n = s_0, s_1, ..., s_n$. Also t_n^+ will refer to the content of t_n, i.e. $t_n^+ = \{s_0, ..., s_n\}$. We will write (Σ^*) for the space of all finite and infinite sequences from Σ^*.

In all that follows, we assume a fixed underlying alphabet Σ. Note that here we will only be concerned with the case of learning from text.

We will be concerned with the learnability of indexable families of uniformly recursive languages, defined as follows.

Definition 1 *Let $\mathcal{C}$ denote a class of non-empty languages. $\mathcal{L} = L_1, L_2, \ldots$ is said to be an* indexing *of $\mathcal{C}$ (written $\mathcal{L} \in Index(\mathcal{C})$) iff $\mathcal{C} = \{L_j \mid j \in \mathbb{N}\}$ and there is a total recursive function p over $\mathbb{N} \times \Sigma^*$ such that, for all $j \in \mathbb{N}$ and $s \in \Sigma^*$, $p(j, s) = 1$ if and only if $s \in L_j$.*

A class $\mathcal{C}$ of non-empty languages is said to be an indexable family *iff there exists an indexing of $\mathcal{C}$.*

We will usually write a class $\mathcal{C}$ as a hypothesis space $H_1, H_2, \ldots$ by which we mean a particular indexing $\mathcal{L}$ of $\mathcal{C}$ where each hypothesis H_i is typically a characteristic function for some $L \in \mathcal{C}$ (when i is called an $\mathcal{L}$-index for L). We will blur the distinction between languages and their characteristic functions, and will write $H_i = L$ if H_i is a characteristic function for L and $H_i = H_j$ if H_i and H_j are characteristic functions for the same $L \in \mathcal{C}$.

We will be concerned with the relationship between data streams (here texts) and underlying concepts (here languages).

Definition 2 *Let L be a language. We say that a finite initial segment t_n of a text $t = s_0, s_1, \ldots$* refutes *L if $(\exists x \leq n) s_x \notin L$.*

Note that t_n refutes L iff there exists no text for L containing t_n as an initial segment.

Following Gold [Go67] we define an *inductive inference machine* (abbr. IIM) to be a Turing machine working as follows. The IIM takes as its input larger and larger initial segments of a text t and it either requests the next input string, or it outputs a hypothesis, i.e. a positive integer which will be interpreted with respect to some underlying indexing $\mathcal{L}$ of the target family $\mathcal{C}$.

A sequence $(j_x)_{x \in \mathbb{N}}$ of numbers is said to be convergent in the limit iff there is a number j such that $j_x = j$ for almost all numbers x.

Now we define some concepts of learning. We start with learning in the limit.

Definition 3 (Go67) *Let $\mathcal{C}$ be a class, $\mathcal{L} = (L_j)_{j \in \mathbb{N}} \in Index(\mathcal{C})$, and $L \in \mathcal{C}$. An IIM $\mathcal{M}$ LIM-TXT-identifies L w.r.t. $\mathcal{L}$ iff on every text t for L $\mathcal{M}$ almost always outputs a hypothesis and the sequence $(\mathcal{M}(t_x))_{x \in \mathbb{N}}$ converges in the limit to a number j such that $L = L_j$.*

An IIM $\mathcal{M}$ LIM-TXT-identifies $\mathcal{C}$ w.r.t. $\mathcal{L}$ iff M LIM-TXT-identifies every $L \in \mathcal{C}$ w.r.t. $\mathcal{L}$.

Let LIM-TXT denote the collection of all $\mathcal{C}$ such that there exists $\mathcal{L} \in Index(\mathcal{C})$ and an IIM $\mathcal{M}$ LIM-TXT-identifying $\mathcal{C}$ w.r.t. $\mathcal{L}$.

We regard this form of identification and its variants as varieties of learning, and indeed use the terms *infer* and *learn* as synonyms for identify.

Note that our learner uses the *sequence* t_m as its input. If the natural restriction is made that the learner's behaviour should be independent of changes in the order of the sequence and the number of repetitions, we have set-driven learning.

Definition 4 (WC80) *An IIM is said to be* set-driven *iff its output depends only on the range of its input, i.e. on any two texts* t, u *we have*

$$(\forall x, y)[t_x^+ = u_y^+ \Rightarrow \mathcal{M}(t_x) = \mathcal{M}(u_y)]$$

We prefix the name of an identification criterion by s- *if in addition we require the learner to be set-driven, e.g. s-LIM-TXT, etc.*

An alternative form of learning is behaviourally-correct learning, defined as follows.

Definition 5 (OW82, CL82) *Let* $\mathcal{C}$ *be a class,* $\mathcal{L} = (L_j)_{j\in\mathbb{N}} \in Index(\mathcal{C})$, *and* $L \in \mathcal{C}$. *An IIM* $\mathcal{M}$ *BC-TXT-identifies* L *w.r.t.* $\mathcal{L}$ *iff on every text* t *for* L $\mathcal{M}$ *almost always outputs a hypothesis and almost every element in the sequence* $(\mathcal{M}(t_x))_{x\in\mathbb{N}}$ *is an index for* L.

An IIM $\mathcal{M}$ *BC-TXT-identifies* $\mathcal{C}$ *w.r.t.* $\mathcal{L}$ *iff* $\mathcal{M}$ *BC-TXT-identifies every* $L \in \mathcal{C}$ *w.r.t.* $\mathcal{L}$.

Let BC-TXT denote the collection of all $\mathcal{C}$ *such that there exists* $\mathcal{L} \in Index(\mathcal{C})$ *and an IIM* $\mathcal{M}$ *BC-TXT-identifying* $\mathcal{C}$ *w.r.t.* $\mathcal{L}$.

Note that, in general, it is undecidable whether or not an IIM has already successfully finished its learning task. If this is decidable, then we obtain finite learning.

Definition 6 (Go67) *Let* $\mathcal{C}$ *be a class,* $\mathcal{L} = L_1, L_2, ... \in Index(\mathcal{C})$, *and* $L \in \mathcal{C}$. *An IIM* $\mathcal{M}$ *FIN-TXT-identifies* L *w.r.t.* $\mathcal{L}$ *iff on every text* t *for* L $\mathcal{M}$ *outputs only a single hypothesis* j *which is an* $\mathcal{L}$*-index for* L, *and stops.*

An IIM $\mathcal{M}$ *FIN-TXT-identifies* $\mathcal{C}$ *w.r.t.* $\mathcal{L}$ *iff* $\mathcal{M}$ *FIN-TXT-identifies every* $L \in \mathcal{C}$ *w.r.t.* $\mathcal{L}$.

Let FIN-TXT denote the collection of all $\mathcal{C}$ *such that there exists* $\mathcal{L} \in Index(\mathcal{C})$ *and an IIM* $\mathcal{M}$ *FIN-TXT-identifying* $\mathcal{C}$ *w.r.t.* $\mathcal{L}$.

A natural property of learning is that the learner should not change its mind without good reason.

Definition 7 (An80) *Let* $\mathcal{C}$ *be a class* $\mathcal{L} = L_1, L_2, ... \in Index(\mathcal{C})$, *and* $L \in \mathcal{C}$. *An IIM* $\mathcal{M}$ *CONSERV-TXT-identifies* L *w.r.t.* $\mathcal{L}$ *iff on every text* t *for* L $\mathcal{M}$ *learns* $\mathcal{L}$ *in the limit and for all* n *if* $\mathcal{M}(t_n) = j$ *is defined then* $\mathcal{M}(t_{n+1}) = j \vee t_{n+1}$ *refutes* L_j.

An IIM $\mathcal{M}$ CONSERV-TXT-identifies $\mathcal{C}$ w.r.t. $\mathcal{L}$ iff $\mathcal{M}$ CONSERV-TXT-identifies every $L \in \mathcal{C}$ w.r.t. $\mathcal{L}$.

Let CONSERV-TXT denote the collection of all $\mathcal{C}$ such that there exists $\mathcal{L} \in Index(\mathcal{C})$ and an IIM $\mathcal{M}$ CONSERV-TXT-identifying $\mathcal{C}$ w.r.t. $\mathcal{L}$.

Various forms of monotonicity requirements on the learner, i.e. that the learner should in some sense output increasingly 'good' hypotheses, are also known.

Definition 8 (Ja91, Wi91) *Let $\mathcal{C}$ be a class, $\mathcal{L} = L_1, L_2, ... \in Index(\mathcal{C})$, and $L \in \mathcal{C}$. An IIM $\mathcal{M}$ identifies L*

1. *strong monotonically*
2. *monotonically*
3. *weak monotonically*

w.r.t. $\mathcal{L}$ iff on every text t for L $\mathcal{M}$ learns L in the limit and if $i_1, i_2, ...$ is the sequence of hypotheses output by $\mathcal{M}$ in learning L from t then

1. $(\forall n) L_{i_n} \subseteq L_{i_{n+1}}$
2. $(\forall n) L_{i_n} \cap L \subseteq L_{i_{n+1}} \cap L$
3. $(\forall n)[t^+_{n+1} \subseteq L_{\mathcal{M}(t_n)} \Rightarrow L_{\mathcal{M}(t_n)} \subseteq L_{\mathcal{M}(t_{n+1})}]$

respectively.

We denote by SMON-TXT, MON-TXT and WMON-TXT those collections of classes $\mathcal{C}$ for which there exists $\mathcal{L} \in Index(\mathcal{C})$ and a learner $\mathcal{M}$ which learns every member L of $\mathcal{C}$ strong monotonically, monotonically, and weak monotonically w.r.t. $\mathcal{L}$, respectively.

We will not concern ourselves with *WMON-TXT* or *MON-TXT* in this paper.

There exists the possibility that the learner may be able to recognise that the text which it is being fed does not represent a text for any language in $\mathcal{C}$, the class which it is trying to learn. Its behaviour in this case should be to output a special symbol $\perp$ 'refuting' the class; otherwise it should learn the class in the limit.

Definition 9 (MA93) *A* refuting inductive inference machine *(RIIM) is a Turing Machine that on any input either behaves like an IIM or outputs the symbol $\perp$ and immediately halts.*

Definition 10 (LW94) *Let t be a text for any language. t is called an* unrepresentative *text for $\mathcal{C}$ if there exists n such that $(\forall L \in \mathcal{C}) t_n$ refutes L. The least such n is called the refutation point of t for $\mathcal{C}$, written $ref(t, \mathcal{C})$.*

Let $\mathcal{C}$ be a class and $\mathcal{L} = L_1, L_2, ... \in Index(\mathcal{C})$. A RIIM $\mathcal{M}$ JREF-TXT-identifies $\mathcal{L}$ iff on any text t for $L \in \mathcal{C}$ $\mathcal{M}$ identifies L in the limit and for all unrepresentative texts t for $\mathcal{C}$ we have $(\exists m \geq ref(t, \mathcal{C}))\mathcal{M}(t_m) = \perp$.

We write $\mathcal{C} \in$ JREF-TXT if there exists $\mathcal{L} \in Index(\mathcal{C})$ and an IIM $\mathcal{M}$ which JREF-TXT-identifies $\mathcal{C}$ w.r.t. $\mathcal{L}$.

3 Popper's Logic of Scientific Discovery - a Précis

In this section we will summarise as much of Popper's philosophical system as we need for our purposes. Even this is quite a task, as this was the major achievement of Popper's professional life and extended to two books [Po34, Po63], and a large number of published papers.

Before Popper the philosophy of science could trace an unbroken line of development back to Bacon. The dominant school, inductivism, held that scientific ideas are gradually proved inductively, by experience - when the idea in question has passed a large number of tests, it may be regarded as effectively proved.

Einstein's overthrow of Newtonian mechanics in the early Twentieth Century provided the intellectual background for Popper's work. If such an established system of scientific law could be disproved[1] then it must have seemed that no scientific idea could ever truly be proved; so indeed Popper reasoned.

Popper built his philosophy of science rigorously from the ground up. He postulated that scientific theories have the character of 'all-statements'; they attempt precisely to specify behaviour of all entities of a certain kind in all circumstances of a certain kind: for example, all planets in rotation about a star. Further, the observations of which empirical science is capable are of a different character; they observe the behaviour of individual entities in individual circumstances.

Popper's first key contribution was to note the asymmetry which arises from this: no number of observations is sufficient to exhaust all the possibilities of an all-statement, even if all these observations are in accord with the predictions of the theory. By contrast, a single observation (allowing for the usual caveats of reliability and inter-subjective repeatability) is enough to refute a theory once and for all, if it conflicts with that theory's predictions. While the theory may be correct in some circumstances, and a useful approximation in others, it does not provide the ultimate, precise truth to which science aspires. An inescapable consequence of this is that scientific theories are never truly proven by observations, for among those observations never, or not yet taken, may be one that disproves the theory.

[1] Newton's laws of course remain useful approximations for many practical purposes.

This demolition of inductivism raises other problems. It was certainly not Popper's intention to suggest that we should stop doing science; but if no theory can be proven, then what may we rationally believe? Popper's answer to this problem forms the starting point for our work.

Those observations which do not refute a particular theory nevertheless play the important role of *corroborating* that theory. Each observation, particularly those which are decisive between theories in the sense that they refute some while corroborating others, may be seen as a test of these theories. When a theory has survived a number of such tests without being refuted, we may say it is well corroborated (though not immune to later refutation) and we may tentatively believe it, for now. It is a small step to Popper's dictum that we should believe the best corroborated theory at any particular time.

Popper formalised the idea of corroboration further by equating the *corroborability* of a theory with its *content*, or *scientific interest*, and further with its logical *falsifiability*. This will be a key idea for us: a theory which has a large number of potential falsifiers (refuting observations) is also potentially more strongly corroborable (in the case that none of these behaviours is ever observed) than a theory with fewer falsifiers.

Popper states in [Po34] (p.395 - all page references to [Po34] are to the 1997 Routledge edition) that:

> I believe that these two ideas - *content* and *degree of corroboration* - are the most important logical tools developed in my book. (Emphasis in original)

In Section 4.1 we will look in detail at Popper's formulation of the degree to which data corroborates a theory, prior to formulating our own laws of corroboration for use in the more restricted field of machine inductive inference.

4 Degree of Corroboration

4.1 Popper's Definition

4.1.1 Discussion

In [Po34] (also [Po54]), Popper went some way towards formalising his key idea of the corroboration lent by examples (or theory) y to theory (synonymously concept or hypothesis) x, calling it $C(x, y)$. We will further formalise the definition of $C(x, y)$, while modifying or discarding some features where necessitated in the light of the following discussion.

It must be mentioned that in [Po34] Popper ties his definition $C(x, y)$ rather rigidly to the notion of absolute logical probability, which has certain unhelpful consequences for our purposes. This is largely because both in [Po34] and in [Po54] he was concerned to *distinguish* his idea of degree of corroboration from any probabilistic definition and needed to demonstrate that to impose the laws of probability on $C(x, y)$ leads to a contradiction. In later work [Po57] he accepted criticism by various authors of this linkage and loosened the definition of $C(x, y)$ accordingly.

Popper remarks [Po54] that

> The particular way in which $C(x, y)$ is here defined I consider unimportant. What may be important are the *desiderata*, and *the fact that they can be satisfied together.* (Emphasis in original)

We will take this as licence to define a function, $c(x, y)$, which differs in some small ways from Popper's $C(x, y)$, while satisfying his desiderata as far as possible. In the next two sections we present first Popper's desiderata for a corroboration function, then our own version, and discuss the differences between them.

4.1.2 Popper's Desiderata

In [Po54] Popper lists nine points which should be satisfied by a corroboration function, and he adds a further one in [Po57].

$C(x, y)$ is in all cases the degree to which y supports or corroborates x, $C(x)$ is the maximum degree to which x may be corroborated, while $P(x)$ is the logical probability of x and $P(x, y)$ is the logical probablity of x given y. $E(x, y)$ is the explanatory power of x with respect to y, and its value is defined based on $P(x), P(y), P(x, y)$ and $P(y, x)$ - we will not be much concerned with this concept. Finally $\overline{x}$ is the logical negation of x.

Popper's desiderata as stated in [Po54] and [Po57] are as follows.

1. $C(x, y)$ is respectively greater than, equal to, or less than 0 iff y supports x, is independent of x, or undermines x.
2. $-1 = C(\overline{y}, y) \leq C(x, y) \leq C(x, x) \leq 1$
3. $0 \leq C(x, x) = C(x) = P(\overline{x}) \leq 1$
4. If y entails x then $C(x, y) = C(x, x) = C(x)$
5. If y entails $\overline{x}$ then $C(x, y) = C(\overline{y}, y) = -1$
6. Let x have a high content, so that $C(x, y)$ approaches $E(x, y)$, and let y support x. Then for any given y, $C(x, y)$ increases with the power of x to explain y (i.e. to explain more and more of the content of y and therefore with the scientific interest of x).
7. If $C(x) = C(y) \neq -1$ then $C(x, u)$ is respectively greater than, equal to or less than $C(y, w)$ whenever $P(x, u)$ is greater than, equal to, or less than $P(y, w)$.

8. If x entails y then: *(a)* $C(x,y) \geq 0$; *(b)* for any given x, $C(x,y)$ and $C(y)$ increase together; and *(c)* for any given y, $C(x,y)$ and $P(x)$ increase together.
9. If $\overline{x}$ is consistent and entails y, then *(a)* $C(x,y) \leq 0$; *(b)* for any given x, $C(x,y)$ and $P(y)$ increase together; *(c)* for any given y, $C(x,y)$ and $P(x)$ increase together.
10. If x is confirmed, supported or corroborated by y so that $C(x,y) \geq 0$, then *(a)* $\overline{x}$ is always undermined by y, i.e. $C(\overline{x},y) < 0$, and *(b)* x is always undermined by $\overline{y}$, i.e. $C(x,\overline{y}) < 0$.

4.2 Our Differences from Popper's Approach - Discussion

4.2.1 Restricted Domain

We wish to define a corroboration function analogous to Popper's but for use in the domain of inductive learning theory. This restricted domain enables us to make a number of simplifying assumptions compared to Popper's version above.

First we note that we always wish to state how well a *hypothesis* is corroborated by *data*. This is already more specific than Popper's approach, in which he specifically allows the corroboration of, for example, one theory by another. Our hypotheses will be those of an *inductive learning machine* (see Section 2) and will come from a particular *hypothesis space*, within which we aim to find a true description of the phenomenon producing the data, which will be a recursive language. The data will be a sequence of *examples* forming a *text* (or strictly speaking, an initial segment of a text) for the phenomenon. To distinguish our corroboration functions from Popper's, we will use lower case. Thus $c(H,t)$ will be the degree to which example text t corroborates hypothesis H.

4.2.2 Fixed Values

Now that we distinguish between theory and data, we are able to simplify further. We assume that data is free of noise, and that we aim to find a hypothesis which *exactly* describes or explains the concept producing the data. Now the idea that data *undermines* (Popper's choice of word) a theory can be replaced by outright refutation in the case that data disagrees with the predictions of the theory. Thus all the possible negative values in Popper's scheme may be replaced in ours by -1, the corroboration value of refuted hypotheses.

Similarly the value 0, reserved by Popper for the degree of corroboration offered to x by an independent theory y, subtly changes its meaning when we restrict ourselves to corroboration of hypotheses by data. The value 0 is now the corroboration given to any theory

- by the empty data set $\emptyset$
- by vacuous data which gives us no help in choosing between competing hypotheses in our space
- in the case that the theory itself is tautological, metaphysical or otherwise not logically refutable.

4.2.3 References to Probability

For historical reasons, Popper's desiderata are tied closely to definitions in probability; specifically, Popper sets out to demonstrate that degree of corroboration is in no sense a measure of probability. For our purposes, we have no need of any directly defined probabilistic measures and so we are able to drop references to $P(x), P(x,y), E(x,y)$, etc. We continue to use $c(H)$ to mean the highest degree of corroboration of which H is capable; however we drop the reference to $P(\overline{x})$ in the definition of $C(x)$ and instead add some natural restrictions on $c(H)$.

Popper's dependence on probabilistic definitions leads him to restrict the maximum degree of corroboration in any case to the value 1. Objections to this unnecessary restriction led him to drop it in [Po57], and we do likewise. Further, we may drop the restriction of degrees of corroboration to real number values altogether, and use any partially ordered set S with a minimum element -1 such that $S - \{-1\}$ has a minimum element 0 and decidable (recursive) relations $\geq, \leq$ and $\bowtie$.

These points having been made, we proceed to our own desiderata.

4.3 Our Definition of Degree of Corroboration

Let H range over hypotheses from our space $\mathcal{L}$, and t over texts and finite initial segments of texts. We assume that $c(H,t)$ ranges over some partially ordered set S with minimum element -1 and an element 0 minimal in $S - \{-1\}$. Similar to Popper, we use $c(H)$ as shorthand for $c(H,H)$, the maximum degree of corroboration possible for H. Falsifiers(H) is the set of potential data items in Σ^* which refute H, and we write $H = H'$ in the case that H and H' describe the same concept.

First we formally define our corroboration functions.

Definition 11 *A corroboration function $c : \mathcal{L} \times (\Sigma^*) \to S$ over $\mathcal{L}$ maps hypotheses and texts to some set S with minimum element -1 and an element 0 minimal in $S - \{-1\}$ such that S has a decidable partial ordering $\leq$, and satisfies the following desiderata for all hypotheses $H, H' \in \mathcal{L}$ and all texts $t, t' \in (\Sigma^*)$:*

1. *$c(H,t) = -1$ iff there exists data in t which refutes H.*
2. *$c(H,t) \geq 0$ iff t does not refute H*

3. *$c(H,t) = 0$ if t is empty or contains no data capable of refutation of any hypothesis in our space.*
4. *$c(H) = max\{Lim_{n\to\infty} c(H,t_n) \mid t$ is a text for $H\}$*
5. *$c(H) \geq c(H')$ if Falsifiers$(H') \subseteq$ Falsifiers(H)*
6. *If t is a finite initial subsequence of t' then either $c(H,t) \leq c(H,t')$ or $c(H,t') = -1$*

Our definition of degree of corroboration is simpler than Popper's because we have dropped all reference to probability and this gives us greater freedom when actually assigning values to our functions $c(H)$ and $c(H,t)$. We will see in the next section that certain inductive learning identification criteria will require corroboration functions with additional properties to those specified above.

Our first three points come from Popper's first four and tenth desiderata. Our fourth and fifth points capture Popper's sense that a high degree of refutability and a high degree of corroborability are synonymous. Our sixth point captures the natural expectation that degree of corroboration of H cannot be decreased by further non-refuting examples (although these same examples may cause an alternative hypothesis H' to become better corroborated than H).

5 Learning with Corroboration

In this section we cover the remaining assumptions and definitions necessary to define a theory of inductive learning with corroboration.

5.1 Hypotheses and Hypothesis Spaces

If two hypotheses describe the same concept, we will write $H_i = H_j$. Note that this is exactly the case Falsifiers(H_i) = Falsifiers(H_j) and may be treated as a shorthand for the latter. If Falsifiers$(H_i) \subseteq$ Falsifiers(H_j) then we will write $H_j \subseteq H_i$ to capture the natural Popperian sense that H_j is more easily refuted (potentially more strongly corroborable) that H_i. None of these relations is necessarily recursive.

We will restrict our attention to *class-preserving* hypothesis spaces, i.e. those indexed recursive families $H_1, H_2, ...$ for $\mathcal{C}$ such that for every $L \in \mathcal{C}$ there exists at least one (and possibly many) i such that $L = H_i$.

Our model of learning requires that $c(H,t)$ and comparison ($\leq$) between degrees of corroboration are both recursive, but not necessarily that $c(H)$ is recursive. Recursiveness of $c(H)$ leads to decidability of $H_i \subseteq H_j$ and therefore of $H_i = H_j$.

All forms of inductive inference suffer from the problem that the learner is required to choose one from among (typically) infinitely many hypotheses at each stage. Clearly no learner can consider all these hypotheses before it outputs a hypothesis or requests further data, so in effect there are only a limited number of hypotheses *in play* at any given time. Most authors gloss over this question as a matter of detail, or deal with it implicitly, but as we intend to propose a new unifying model for machine inductive inference, we feel constrained to deal with it explicitly.

We therefore assume that along with our hypothesis space $H_1, H_2, ...$ we have a recursive, monotonically increasing function $ip : \mathbb{N} \rightarrow \mathbb{N}$ with $Lim_{n\rightarrow\infty} ip(n) = \infty$ which gives the number of hypotheses in play at stage n of any learning procedure with this hypothesis space. This leads to one slight concession with respect to our desiderata: hypotheses H_j which are not yet in play at stage n need not be considered to be either refuted or corroborated by t_n, the examples seen to that stage - we therefore arbitrarily assign $c(H_j, t_n) = 0$ for such n, j. This cannot cause confusion as these hypotheses are (by definition) not considered by any algorithm; it serves only to simplify some algorithms defined later.

5.2 Corroboration Functions and Canonical Learners with Corroboration

In the following section (Section 6) we examine the use of corroboration in inductive learning and prove that many of the most natural inductive learning identification types can be characterised by an existence condition for a suitable corroboration function over the hypothesis space. Our intention is that this corroboration function (which is invariably recursive so no undecidability results are implied, nor is any additional computing power gained illicitly) will be used as an oracle by a canonical learner for the appropriate type; this demonstrates that there is effectively a single best learning strategy for each identification type, and only the details of the corroboration function change depending on the hypothesis space.

The behaviour of a learner with corroboration is defined as follows.

Definition 12 *Turing machine $\mathcal{M}$, with oracle $c(H, t)$ is called a* learner with corroboration *if $c(H, t)$ is a recursive corroboration function and on input t with hypotheses $H_1, .., H_p$ in play, $\mathcal{M}$ outputs some $i \leq p$ such that $c(H_i, t) > 0$ is maximal among the $c(H_j, t), j = 1, ..., p$, if defined, and requests more input otherwise.*

If additionally $\mathcal{M}$ learns within identification type $*$, we call $\mathcal{M}$ a $*$-learner with corroboration.

Clearly such a learner is consistent with Popper's dictum that we should prefer the most strongly corroborated hypothesis among competing hypotheses.

6 Characterising TXT-Identification Types in Learning with Corroboration

In this section we are concerned only with learning from text, and often abbreviate the names of identification types by dropping the -*TXT*.

6.1 BC- and LIM-learning

Definition 13 *A corroboration function c over $\mathcal{L}$ is called* cycling *iff*

$$(\forall H \in \mathcal{L})(\forall t \in Txt(H))(\exists n)(\exists D \subseteq \mathbb{N})[(\forall i \in D)H_i = H \wedge$$
$$(\forall m \geq n)(\exists i \in D)(\forall j)[c(H_i, t_m) > c(H_j, t_m) \vee [c(H_i, t_m) \not< c(H_j, t_m) \wedge i \leq j]]]$$

Theorem 1 *$\mathcal{C} \in BC\text{-}TXT$ iff there exists $\mathcal{L} \in Index(\mathcal{C})$ such that there is a recursive cycling corroboration function c over $\mathcal{L}$.*

Proof

($\Leftarrow$)

We define a learner $\mathcal{M}$ which uses such a recursive cycling c to BC-learn any $H \in \mathcal{L}$.

Let t be a text for $H \in \mathcal{L}$. Let the hypotheses in play at stage m be $H_1, ..., H_p$. At the mth stage (i.e. on input t_m) $\mathcal{M}$ behaves as follows.

$$\mathcal{M}(t_m) \begin{cases} = min(Best_m) & \text{if defined} \\ \text{requests more input} & \text{otherwise} \end{cases}$$

where

$$\text{Best}_m = \{i \mid i \leq p \wedge c(H_i, t_m) > 0 \wedge (\forall j \leq p) c(H_i, t_m) \not< c(H_j, t_m)\}$$

$\mathcal{M}$ is recursive: $\mathcal{M}$ recursively computes $c(H_i, t_m)$ for $i = 1, ..., p$ and forms the finite set of those i for which $c(H_i, t_m) > 0$ is maximal under the recursive relation $\leq$. $\mathcal{M}$ now outputs the minimum such i, unless the set is empty, in which case it requests more input.

On presentation of a text for $H \in \mathcal{C}$ there exists a stage n after which $\mathcal{M}$ always outputs an $\mathcal{L}$-index for H: let t be a text for H. By assumption, c is a cycling corroboration function, so there exists a set D such that $(\forall i \in D)H_i = H$ and a stage n such that $(\forall m \geq n) min(Best_m) \in D$. The result follows from the definition of $\mathcal{M}$.

($\Rightarrow$)

Suppose $\mathcal{M}$ is an inductive learning machine which BC-learns $\mathcal{C}$ w.r.t. $\mathcal{L} = H_1, H_2, ...$ and let t be a text. We define a recursive c which produces values (for degree of corroboration) ranging over $\mathbb{N} \cup \{-1\}$. Let

$$c(H_i, t_m) = \begin{cases} -1 & \text{if } t_m \text{ refutes } H_i \\ \mid t_m \mid +1 & \text{if } \mathcal{M}(t_m) = i \\ \mid t_m \mid & \text{otherwise} \end{cases}$$

c is recursive: it is decidable for any i whether t_m refutes H_i, and by assumption $\mathcal{M}$ is a Turing machine which always outputs a hypothesis or requests further input.

c is a cycling corroboration function over $\mathcal{L}$: let t be a text for $H \in \mathcal{C}$. By assumption there exists a set D such that for all $i \in D$ we have $H_i = H$, and a stage n such that for all $m \geq n$ our learner $\mathcal{M}$ outputs an index i such that $i \in D$. Therefore at all stages $m \geq n$ there always exists an $i \in D$ such that $c(H_i, t_m) > c(H_j, t_m)$ for all $j \neq i$, which satisfies the requirements of Definition 13.

□

Corollary 1 *If $\mathcal{C} \in BC\text{-}TXT$ then there exists $\mathcal{L} \in Index(\mathcal{C})$ and a recursive cycling corroboration function c over $\mathcal{L}$ with the property that*

$$(\forall H \in \mathcal{L})(\forall t \in Txt(H))(\exists n)(\exists D \subseteq \mathbb{N})[(\forall i \in D) H_i = H \wedge \\ (\forall m \geq n)(\exists i \in D)(\forall j \neq i) c(H_i, t_m) > c(H_j, t_m)]$$

Proof

Immediate from proof of Theorem 1 ($\Rightarrow$).

□

Corollary 2 *There is a canonical BC-learner with corroboration which will learn any $\mathcal{C} \in BC\text{-}TXT$ w.r.t. any $\mathcal{L} \in Index(\mathcal{C})$ using any recursive cycling corroboration function c over $\mathcal{L}$ as an oracle.*

Proof

Immediate from the $\Leftarrow$ direction of the proof of Theorem 1 as the definition of $\mathcal{M}$ does not depend on $\mathcal{C}$ except via c.

□

Definition 14 *A corroboration function c over $\mathcal{L}$ is called* limiting *iff*

$$(\forall H \in \mathcal{L})(\forall t \in Txt(H))(\exists i)[H_i = H \wedge \\ (\exists n)(\forall m \geq n)(\forall j)[c(H_i, t_m) > c(H_j, t_m) \vee [c(H_i, t_m) \not< c(H_j, t_m) \wedge i \leq j]]]$$

Clearly a limiting corroboration function is also a cycling corroboration function with $\mid D \mid = 1$.

Theorem 2 *$\mathcal{C} \in LIM\text{-}TXT$ iff there exists $\mathcal{L} \in Index(\mathcal{C})$ such that there is a recursive limiting corroboration function c over $\mathcal{L}$.*

Proof

($\Leftarrow$)

We define a learner $\mathcal{M}$ which uses such a recursive limiting c to *LIM*-learn any $H \in \mathcal{L}$.

Let t be a text. Let the hypotheses in play at stage m be $H_1, ..., H_p$. At the mth stage (i.e. on input t_m) $\mathcal{M}$ behaves as follows.

$$\mathcal{M}(t_m) \begin{cases} = min(Best_m) & \text{if defined} \\ \text{requests more input} & \text{otherwise} \end{cases}$$

where

$$\text{Best}_m = \{i \mid i \leq p \wedge c(H_i, t_m) > 0 \wedge (\forall j \leq p) c(H_i, t_m) \not< c(H_j, t_m)\}$$

$\mathcal{M}$ is recursive: $\mathcal{M}$ recursively computes $c(H_i, t_m)$ for $i = 1, ..., p$ and forms the finite set of those i for which $c(H_i, t_m)$ is maximal under the recursive relation $\leq$. $\mathcal{M}$ now outputs the minimum such i, unless the set is empty, in which case it requests more input.

On presentation of a text t for H, $\mathcal{M}$ converges to some j such that $H_j = H$: fix t, an arbitrary text for H. Let n be that stage defined in Definition 14. Now there is some j with $H_j = H$ such that at stage n and all subsequent stages m $\mathcal{M}$ will output j because $j = min(Best_m)$ by assumption that c is a limiting corroboration function and the definition of $\mathcal{M}$.

($\Rightarrow$)

Suppose $\mathcal{M}$ is an inductive learning machine which *LIM*-learns $\mathcal{C}$ w.r.t. $\mathcal{L}$. We define a recursive c which produces values (for degree of corroboration) ranging over $\mathbb{N} \cup \{-1\}$. Let

$$c(H_j, t_m) = \begin{cases} -1 & \text{if } t_m \text{ refutes } H_j \\ \mid t_m \mid +1 & \text{if } \mathcal{M}(t_m) = j \\ \mid t_m \mid & \text{otherwise} \end{cases}$$

c is recursive: it is decidable for any j, whether t_m refutes H_j, and by assumption $\mathcal{M}$ is an IIM.

c is a limiting corroboration function over $\mathcal{L}$: let t be any text for $H \in \mathcal{C}$. By assumption there exists an index j such that $H_j = H$ and a stage n after which $\mathcal{M}$ always outputs j. Therefore at all stages $m \geq n$ we have $c(H_j, t_m) > c(H_k, t_m)$ for all $k \neq j$, which satisfies the requirements of Definition 14.

□

Corollary 3 *If $\mathcal{C} \in LIM\text{-}TXT$ then there exists $\mathcal{L} \in Index(\mathcal{C})$ such that there is a recursive limiting corroboration function c over $\mathcal{L}$ with the property that*

$$(\forall H \in \mathcal{L})(\forall t \in Txt(H))(\exists i)[H_i = H \wedge (\exists n)(\forall m \geq n)(\forall j \neq i) c(H_i, t_m) > c(H_j, t_m)]$$

Proof

Immediate from proof of Theorem 2 ($\Rightarrow$).

$\square$

Corollary 4 *There is a canonical LIM-learner with corroboration which will learn any $\mathcal{C} \in$ LIM-TXT w.r.t. any $\mathcal{L} \in Index(\mathcal{C})$ using any recursive limiting corroboration function c over $\mathcal{L}$ as an oracle.*

Proof

Immediate from the $\Leftarrow$ direction of the proof of Theorem 2 as the definition of $\mathcal{M}$ does not depend on $\mathcal{C}$ except via c.

$\square$

Corollary 5 *There is a canonical (LIM$\cup$BC)-learner with corroboration which will BC-learn any $\mathcal{C} \in$ BC-TXT w.r.t. any $\mathcal{L} \in Index(\mathcal{C})$ using any recursive cycling corroboration function c over $\mathcal{L}$ as an oracle and will LIM-learn any $\mathcal{C} \in$ LIM-TXT w.r.t. any $\mathcal{L} \in Index(\mathcal{C})$ using any recursive limiting corroboration function c over $\mathcal{L}$ as an oracle.*

Proof

The same canonical learner is used in Corollaries 2 and 4.

$\square$

Our approach of learning with corroboration allows us to prove the known result (it appears to be a 'folk theorem') that *BC-TXT* = *LIM-TXT* as follows.

Theorem 3 *BC-TXT* = *LIM-TXT*

Proof

That *LIM-TXT* $\subseteq$ *BC-TXT* is obvious from the definitions.

We show that any learner $\mathcal{M}$ which *BC*-learns $\mathcal{C}$ w.r.t. $\mathcal{L}$ permits the construction of a learner $\mathcal{M}'$ which *LIM*-learns $\mathcal{C}$ w.r.t. $\mathcal{L}$. Our proof method is to build $\mathcal{M}'$ to copy $\mathcal{M}$ until, by enumerating longer and longer initial segments of the characteristic functions for H, the hypothesis of $\mathcal{M}$, and H', the hypothesis of $\mathcal{M}'$, we have proof that $\mathcal{M}$ has 'really' changed its hypothesis, instead of just switching to another hypothesis describing the same language.

Define the *unchanged length* $UL(\mathcal{M}, t_{n+1})$ of a learner $\mathcal{M}$ at stage $n+1$ to be the length of the longest sequence of stages ending in n at which the learner output the same hypothesis.

We define $\mathcal{M}'$ and c using mutual recursion as follows.

$$c(H_i, t_{m+1}) = \begin{cases} -1 & \text{if } t_{m+1} \text{ refutes } H_i \\ \mid t_{m+1} \mid +1 & \text{if } [\mathcal{M}'(t_m) \text{ is undefined} \vee \\ & (\exists j : 0 \leq j \leq UL(\mathcal{M}', t_{m+1}))H_{\mathcal{M}'(t_m)}(j) \neq H_{\mathcal{M}(t_{m+1})}(j)] \\ & \wedge \mathcal{M}(t_{m+1}) = i \\ \mid t_{m+1} \mid +1 & \text{if } \mathcal{M}'(t_m) = i \wedge \\ & (\forall j : 0 \leq j \leq UL(\mathcal{M}', t_{m+1}))H_{\mathcal{M}'(t_m)}(j) = H_{\mathcal{M}(t_{m+1})}(j) \\ \mid t_{m+1} \mid & otherwise \end{cases}$$

The second and third cases above are mutually exclusive in the sense that at most one of these cases will apply to at most one i at any stage. They have been separated for clarity as they represent the cases where $\mathcal{M}'(t_m)$ is defined and $\mathcal{M}'(t_{m+1})$ does not/does equal $\mathcal{M}'(t_m)$, respectively.

We define $\mathcal{M}'$ in the familiar way:

$$\mathcal{M}'(t_m) \begin{cases} = min(Best_m) & \text{if defined} \\ \text{requests more input} & \text{otherwise} \end{cases}$$

where

$$\text{Best}_m = \{i \mid i \leq p \wedge c(H_i, t_m) > 0 \wedge (\forall j \leq p) c(H_i, t_m) \not< c(H_j, t_m)\}$$

Clearly c and $\mathcal{M}'$ are recursive. It is easily checked that if $\mathcal{M}$ BC-learns $\mathcal{L}$ then c is a limiting corroboration function over $\mathcal{L}$ and consequently by Theorem 2 $\mathcal{M}'$ LIM-learns $\mathcal{L}$, as required.

□

6.2 Set-driven learning

When considering the philosophical background for our model of learning, it seems clear that the order in which examples are presented to the learner, or the number of times the same example is repeated, has no significance. This leads us to the following definition.

Definition 15 *A corroboration function c over $\mathcal{L} = H_1, H_2, ...$ is called* natural *if on all texts t, u, for all m, n we have*

$$t_m^+ = u_n^+ \Rightarrow (\forall i) c(H_i, t_m) = c(H_i, u_n)$$

It might be objected that corroboration functions lacking the naturalness property should be disallowed. However, they are no more unnatural than non-set-driven learners (it is known [LZ94] that *s-LIM-TXT* $\subset$ *LIM-TXT*).

Theorem 4 *$\mathcal{C} \in$ s-LIM-TXT iff there exists $\mathcal{L} \in Index(\mathcal{C})$ such that there exists a recursive natural limiting corroboration function c over $\mathcal{L}$.*

Proof

($\Leftarrow$)

Let c be a recursive natural limiting corroboration function over $\mathcal{L}$. Let $\mathcal{M}$ be the learner from the $\Leftarrow$ proof of Theorem 2, in which it has already been shown that $\mathcal{M}$ *LIM*-learns $\mathcal{L}$. We show that $\mathcal{M}$ is set-driven as follows. Let u, t be texts and let $Best_{t,m}$ and $Best_{u,n}$ be defined similarly to $Best_m$ in the $\Leftarrow$ proof of Theorem 2. It is clear that given the naturalness of c we have

$$t_m^+ = u_n^+ \Rightarrow Best_{t,m} = Best_{u,n}$$

and so $t_m^+ = u_n^+ \Rightarrow \mathcal{M}(t_m) = \mathcal{M}(u_n)$ and $\mathcal{C} \in$ *s-LIM-TXT* as required.

($\Rightarrow$)

Let $\mathcal{C} \in$ *s-LIM-TXT* via set-driven learner $\mathcal{M}$ working w.r.t. $\mathcal{L}$. Let c be the corroboration function defined as follows:

$$c(H_j, t_m) = \begin{cases} -1 & \text{if } t_m \text{ refutes } H_j \\ \mid t_m^+ \mid +1 & \text{if } \mathcal{M}(t_m) = j \\ \mid t_m^+ \mid & \text{otherwise} \end{cases}$$

Now let t, u be texts and suppose $t_m^+ = u_n^+$ for some m, n. Now clearly we have $(\forall i) c(H_i, t_m) = c(H_i, u_n)$ as required.

□

Corollary 6 *There is a canonical s-LIM-learner with corroboration which will learn any $\mathcal{C} \in$ s-LIM-TXT w.r.t. any $\mathcal{L} \in Index(\mathcal{C})$ using any recursive natural limiting corroboration function c over $\mathcal{L}$ as an oracle.*

Proof

Immediate from the $\Leftarrow$ direction of the proof of Theorem 4 as the definition of $\mathcal{M}$ does not depend on $\mathcal{C}$ except via c.

□

6.3 Conservative and Strong Monotonic learning

Definition 16 *A corroboration function $c : \mathcal{L} \times (\Sigma^*) \to S$ over $\mathcal{L}$ is called* attaining *if*

$$(\forall H \in \mathcal{L})(\forall t \in Txt(H))[(\exists j)(\exists n)[H_j = H \wedge c(H_j, t_n) \in c'(H_j)] \wedge$$
$$(\forall i)(\forall m)[c(H_i, t_m) \in c'(H_i) \Rightarrow (\forall H' \in \mathcal{L})[t_m \textit{ refutes } H' \vee H_i \not\supset H']]]$$

where

$$c'(H_i) = \{c(H_j) \mid H_j = H_i \wedge (\forall k)[c(H_k) > c(H_j) \Rightarrow H_k \neq H_i]\}$$

c is a recursive attaining corroboration function *if both c and* $c'_f : \mathcal{L} \times S \rightarrow \{0,1\}$ *defined by*

$$c'_f(H_i, s) = \begin{cases} 1 & \text{if } s \in c'(H_i) \\ 0 & \text{otherwise} \end{cases}$$

are total and recursive.

Note that $c(H_i, t_m) \in c'(H_i)$ implies $c(H_i, t_m) = c(H_i)$ and that $c(H, \emptyset) = 0$ implies $(\forall s \in c'(H)) s \geq 0$.

Theorem 5 *$\mathcal{C} \in$ CONSERV-TXT iff there exists $\mathcal{L} \in Index(\mathcal{C})$ such that there exists a recursive attaining corroboration function c over $\mathcal{L}$.*

Proof

($\Leftarrow$)

Let $c(H, t)$ be a recursive, attaining corroboration function over $\mathcal{L}$. We define a *CONSERV*-learner $\mathcal{M}$ on text t as follows.

$$\mathcal{M}(t_m) \begin{cases} = \mathcal{M}(t_{m-1}) & \text{if defined and } (t_m \text{ does not refute } \mathcal{M}(t_{m-1}) \\ & \text{or } Best_m \text{ is undefined}) \\ = min(Best_m) & \text{if defined and } (t_m \text{ refutes } \mathcal{M}(t_{m-1}) \\ & \text{or } \mathcal{M}(t_{m-1}) \text{ is undefined}) \\ \text{requests more input} & \text{otherwise} \end{cases}$$

where

$$\text{Best}_m = \{i \mid i \leq p \wedge c(H_i, t_m) \in c'(H_i) \wedge (\forall j \leq p) c(H_j, t_m) \not> c(H_i, t_m)\}$$

where $H_1, ..., H_p$ are the hypotheses in play at stage m.

$\mathcal{M}$ is recursive: it is not difficult to see that $Best_m$ is a recursive set as its computation involves only finitely many computations of $c(H_i, t_m)$ and $c'_f(H_i, c(H_i, t_m))$, both of which are recursive by assumption. The result follows.

For all m such that $\mathcal{M}(t_m)$ is defined, $\mathcal{M}(t_{m+1}) = \mathcal{M}(t_m)$ or t_{m+1} refutes $H_{\mathcal{M}(t_m)}$: immediate from the definition of $\mathcal{M}$.

On any text t for $H \in \mathcal{L}$, $\mathcal{M}$ converges to some j with $H_j = H$: let j be that index of H (from Definition 16) for which $(\exists n) c(H_j, t_n) \in c'(H_j)$ (and so $(\forall m \geq n) c(H_j, t_m) \in c'(H_j)$).

We show that

$$(\forall k < j)[H_k = H_j = H \vee (\exists m' \geq m)(\forall n \geq m')c(H_k, t_n) \notin c'(H_k)]$$

Suppose $k < j$, $H_k \neq H_j$ and H_k is never refuted by t (if H_k is refuted by $t_{m'}$ then $c(H_k, t_{m'}) = -1 < c(H_j, t_{m'})$ and we are done). Now $(\forall n)t_n^+ \subseteq H_j \subset H_k$ so by Definition 16 we cannot have $c(H_k, t_n) \in c'(H_k)$, and we are done.

Finally we show that

$$(\exists n \geq m')\mathcal{M}(t_n) = j \vee (\forall n \geq m')[\mathcal{M}(t_n) = \mathcal{M}(t_{m'}) \wedge H_{\mathcal{M}(t_{m'})} = H]$$

Suppose $\mathcal{M}(t_{m'}) = k$ and $H_k = H$. Then clearly H_k will never be refuted by t, so by the definition of $\mathcal{M}$ we are done. If $\mathcal{M}(t_{m'-1})$ is undefined, then $min(Best_{m'}) = k \leq j$ with $H_k = H$ from above, so again we are done. The only remaining case is when $\mathcal{M}(t_{m'}) = k$ and $H_k \neq H$. Then at some stage $n \leq m'$ we have $c(H_k, t_n) \in c'(H_k)$ by the definition of $\mathcal{M}$, so $H_k \not\supset H_j$ by Definition 16. Now because t is a text for H, at some stage $n' > m'$ we will have $t_{n'}$ refutes H_k, and $min(Best_{n'}) = k \leq j$ with $H_k = H$ as before so $H_{\mathcal{M}(t_{n'})} = H$. In all cases $\mathcal{M}$ converges on t to an index for H as required.

$(\Rightarrow)$

Let $\mathcal{M}$ be a learner that learns $\mathcal{C}$ conservatively w.r.t. $\mathcal{L}$. We define a recursive attaining corroboration function $c : \mathcal{L} \times (\Sigma^*) \rightarrow \{-1, 0, 1\}$ with $c'(H) = \{1\}$ as follows.

$$c(H_i, t_n) = \begin{cases} -1 & \text{if } t_n \text{ refutes } H_i \\ 1 & \text{if } t_n \text{ does not refute } H_i \wedge \mathcal{M}(t_n) = i \\ 0 & \text{otherwise} \end{cases}$$

$c(H, t)$ *is recursive*: follows immediately from the recursiveness of $\mathcal{M}$ and the decidability of whether t_n refutes H_i.

We now prove the two properties necessary to prove c is an attaining corroboration function (Definition 16). Fix $H \in \mathcal{C}$ and $t \in Txt(H)$.

(i) $(\exists j)(\exists n)[H_j = H \wedge c(H_j, t_n) = 1]$: by assumption there exists some j with $H_j = H$ and $(\exists n)(\forall m \geq n)\mathcal{M}(t_m) = j$. Then by definition of $c(H, t)$, we have $c(H_j, t_n) = 1$ as required. This also suffices to prove that $(\forall i)c'(H_i) = \{1\}$ and so c'_f is total and recursive.

(ii) $(\forall i)(\forall n)[c(H_i, t_n) \in c'(H_i) \Rightarrow (\forall H' \in \mathcal{L})[t_n \text{ refutes } H' \vee H_i \not\supset H']]$: let $c(H_i, t_n) = 1$. Then by definition of $c(H, t)$ we have that $\mathcal{M}(t_n) = i$. By assumption $\mathcal{M}$ *CONSERV*-learns $\mathcal{C}$ w.r.t. $\mathcal{L}$, so there is no j with $t_n^+ \subseteq H_j \subset H_i$ because if there were then we would be able to extend t_n to a text t' for H_j and $\mathcal{M}$ would fail to *CONSERV*-learn H_j on t', a contradiction.

$c(H, t)$ *is an attaining corroboration function over* $\mathcal{L}$: immediate from *(i), (ii)* above.

□

Corollary 7 *There is a canonical CONSERV-learner with corroboration which will learn any $\mathcal{C} \in$ CONSERV-TXT w.r.t. any $\mathcal{L} \in Index(\mathcal{C})$ using as an oracle any recursive attaining corroboration function c over $\mathcal{L}$.*

Proof

Immediate from the $\Leftarrow$ direction of the proof of Theorem 5 as the definition of $\mathcal{M}$ does not depend on $\mathcal{C}$ except via c and c'.

$\square$

Definition 17 *A corroboration function $c(H,t)$ over $\mathcal{L} = H_1, H_2, ...$ is called* strict *if*

$$(\forall H_i \in \mathcal{L})(\forall t \in Txt(H_i))(\forall n)[c(H_i, t_n) \in c'(H_i) \Rightarrow (\forall H_j \supseteq t_n^+)H_j \supseteq H_i]$$

where

$$c'(H) = \{c(H_i) \mid H_i = H \wedge (\forall j)[c(H_j) > c(H_i) \Rightarrow H_j \neq H]\}$$

c is called a recursive strict corroboration function *if both c and $c'_f : \mathcal{L} \times S \rightarrow \{0,1\}$ defined by*

$$c'_f(H_i, s) = \begin{cases} 1 & \text{if } s \in c'(H_i) \\ 0 & \text{otherwise} \end{cases}$$

are total and recursive.

Theorem 6 *$\mathcal{C} \in$ SMON-TXT iff there exists $\mathcal{L} \in Index(\mathcal{C})$ such that there exists a recursive strict attaining corroboration function c over $\mathcal{L}$.*

Proof

($\Leftarrow$) Let $c(H,t)$ be such a function over $\mathcal{L}$. We define $\mathcal{M}$ to *SMON*-learn $\mathcal{C}$ w.r.t. $\mathcal{L}$ as follows.

$$\mathcal{M}(t_m) \begin{cases} = \mathcal{M}(t_{m-1}) & \text{if defined and } (t_m \text{ does not refute } H_{\mathcal{M}(t_{m-1})} \\ & \vee Best_m \text{ is undefined}) \\ = min(Best_m) & \text{if defined and } (t_m \text{ refutes } \mathcal{M}(t_{m-1}) \\ & \text{or } \mathcal{M}(t_{m-1}) \text{ is undefined}) \\ \text{requests more input} & \text{otherwise} \end{cases}$$

where

$$\text{Best}_m = \{i \mid i \leq p \wedge c(H_i, t_m) \in c'(H_i) \wedge (\forall j \leq p) c(H_j, t_m) \not> c(H_i, t_m)\}$$

where $H_1, ..., H_p$ are the hypotheses in play at stage m.

$\mathcal{M}$ *is recursive*: only finitely many recursive computations of $c(H_i, t_m)$ and $c'_f(H_i, c(H_i, t_m))$ are needed on input t_m.

For all stages m at which $\mathcal{M}(t_m)$ is defined, we have $H_{\mathcal{M}(t_m)} \subseteq H_{\mathcal{M}(t_{m+1})}$: let $\mathcal{M}(t_m) = i$. Then in particular t_m refutes all H_j such that $H_j \not\supseteq H_i$, because c is strict. Refuted hypotheses remain refuted, so if $\mathcal{M}(t_{m+1}) = j$ we have $H_j \supseteq H_i$. Finally it is clear from the definition of $\mathcal{M}$ that $\mathcal{M}(t_{m+1})$ is defined.

On any text t for $H \in \mathcal{C}$ $\mathcal{M}$ converges to some j with $H_j = H$: because c is attaining, this is identical to the same part of the proof of Theorem 5 ($\Leftarrow$).

($\Rightarrow$) Let $\mathcal{C} \in$ *SMON-TXT*, and suppose $\mathcal{M}$ is a learner which learns $\mathcal{C}$ strong monotonically w.r.t. $\mathcal{L} = H_1, H_2, \ldots$ We define a recursive, strict, attaining c as required.

$$c(H_i, t_m) = \begin{cases} -1 & \text{if } t_m \text{ refutes } H_i \\ 1 & \text{if } t_m \text{ does not refute } H_i \wedge \mathcal{M}(t_m) = i \\ 0 & \text{otherwise} \end{cases}$$

c is recursive: immediate from the recursiveness of $\mathcal{M}$.

Fix $H \in \mathcal{C}$ and let t be a text for H.

c is attaining: by assumption, there exists a stage m such that $\mathcal{M}(t_m) = i$ for some $H_i = H$. Then $c(H_i, t_m) = 1$. This also proves that $c'(H_i) = \{1\}$ and so c'_f is a total recursive function.

c is strict: suppose for a contradiction that $c(H_i, t_m) \in c'(H_i)$ and there exists j such that $H_j \not\supseteq H_i$ and t_m does not refute H_j. Then we can extend t_m into a text t' for H_j and $\mathcal{M}$ fails to learn H_j strong monotonically on t', contrary to our assumption.

□

Corollary 8 *There exists a canonical SMON-learner with corroboration which SMON-learns any $\mathcal{C} \in$ SMON-TXT w.r.t. any $\mathcal{L} \in Index(\mathcal{C})$ using any recursive strict attaining corroboration function over $\mathcal{L}$ as an oracle.*

Proof

Immediate from the proof of Theorem 6 ($\Leftarrow$).

□

Corollary 9 *There is a canonical (CONSERV$\cup$SMON)-learner with corroboration which will CONSERV-learn any $\mathcal{C} \in$ CONSERV-TXT w.r.t. any $\mathcal{L} \in Index(\mathcal{C})$ using any recursive attaining corroboration function c over $\mathcal{L}$ as an oracle and will SMON-learn any $\mathcal{C} \in$ SMON-TXT w.r.t. any $\mathcal{L} \in Index(\mathcal{C})$ using any recursive strict attaining corroboration function c for $\mathcal{L}$ as an oracle.*

Proof
The same canonical learner is used in Corollaries 7 and 8.
□

It is known [LZ93] that *CONSERV-TXT* = *WMON-TXT* and so immediately from Definition 8 we have the following fact. The learning with corroboration approach allows an alternative proof.

Corollary 10 *SMON-TXT* $\subseteq$ *CONSERV-TXT*

Proof
A necessary and sufficient condition for membership of *SMON-TXT* is the existence of a recursive, strict attaining corroboration function over $\mathcal{L}$ (Theorem 6), which is stronger than the necessary and sufficient condition for membership of *CONSERV-TXT* given in Theorem 5.
□

Strictness of this containment is proved by example [LZ93].

6.4 FIN- and refuting learning

Definition 18 *Let* $\mathcal{L} = H_1, H_2, ...$ *be a hypothesis space. Then* $f : (\Sigma^*) \times \mathbb{N} \to \{0,1\}$ *is called a* sufficiency function *over* $\mathcal{L}$ *if*

$$\begin{array}{l}(\forall t)(\forall m)(\forall n)[f(t_m, n) = 1 \\ \Rightarrow [(\forall j) t_m \textit{ refutes } H_j \vee \\ (\exists i \leq n)[t_m^+ \subseteq H_i \wedge (\forall k)[H_k = H_i \vee t_m \textit{ refutes } H_k]]]]\end{array}$$

and

$$(\forall t)(\forall j)(\forall k \geq j)(\forall n)(\forall m \geq n)[f(t_j, n) = 1 \Rightarrow f(t_k, m) = 1]$$

Definition 19 *Let* f *be a sufficiency function over* $\mathcal{L}$.
f *is called an* inner sufficiency function *over* $\mathcal{L}$ *if it additionally holds that for every text* $t \in Txts(\mathcal{L})$, $(\exists m, n) f(t_m, n) = 1$.
If instead it holds that for every text $t \notin Txts(\mathcal{L})$, $(\exists m, n) f(t_m, n) = 1$, *then* f *is called an* outer sufficiency function *over* $\mathcal{L}$.

Intuitively, a sufficiency function $f(t, n)$ monitors whether there are hypotheses in $\mathcal{L}$ which are not yet in play (i.e. have no index less than or equal to n), and which would not be refuted by E if they were in play. When it returns 1 then this condition has ceased to be true (so we may look for an explanation of t solely in $H_1, ..., H_n$) and further, at most one $H \in \mathcal{L}$ has indices less than or equal to n whose accompanying hypothesis is unrefuted by t.

An inner or outer sufficiency function ensures that (under certain circumstances to do with the limiting behaviour of the data stream) if the condition ceases to be true then 1 will be returned at some later time.

Naturally the existence of a recursive (inner or outer) sufficiency function over $\mathcal{L}$ is a very strong condition and allows particularly strong forms of learning.

Theorem 7 *$\mathcal{C} \in FIN\text{-}TXT$ iff there exists $\mathcal{L} \in Index(\mathcal{C})$ such that there exists a recursive inner sufficiency function over $\mathcal{L}$.*

Proof

($\Leftarrow$)

Suppose the existence of a recursive inner sufficiency function f over $\mathcal{L} = H_1, H_2, ...$ Let t be a text and the hypotheses in play at stage m be $H_1, ..., H_p$. $\mathcal{M}$ behaves as follows.

$$\mathcal{M}(t_m) \begin{cases} = i \text{ (and halt)} & \text{if } f(t_m, p) = 1 \wedge i = min\{j \mid t_m^+ \subseteq H_j \wedge j \leq p\} \\ \text{requests more input} & \text{otherwise} \end{cases}$$

$\mathcal{M}$ is recursive: immediate from the recursiveness of f and the finiteness of t_m.

Fix $H \in \mathcal{C}$ and $t \in Txt(H)$.

$\mathcal{M}$ only ever outputs one hypothesis, which is an $\mathcal{L}$-index for H, then halts: because f is an inner sufficiency function, there exists m such that $f(t_m, p) = 1$ where $H_1, ..., H_p$ are the hypotheses in play at stage m. Then there exists only one $H \in \mathcal{C}$ which has indices $i \leq p$ such that t_m does not refute H_i; $\mathcal{M}$ outputs such an index and halts at stage m.

($\Rightarrow$)

Let $\mathcal{L} = H_1, H_2, ...$ and suppose $\mathcal{M}$ is an inductive learning machine which *LIM*-learns $\mathcal{C}$ w.r.t. $\mathcal{L}$. Let t be a text.

Define $f(t, n)$ as follows.

$$f(t_m, n) = \begin{cases} 1 & \text{if } \mathcal{M}(t_m) = i \leq n \\ 0 & \text{otherwise} \end{cases}$$

f is a sufficiency function over $\mathcal{L}$: suppose $\mathcal{M}$ first outputs i at stage m, when the hypotheses in play are $H_1, .., H_j$. Naturally $j \geq i$ and $t_m^+ \subseteq H_i$. It holds that $\neg(\exists k)[H_k \neq H_i \wedge t_m^+ \subseteq H_k]$ because otherwise we could extend the initial segment t_m to a text t' for H_k and $\mathcal{M}$ would fail to *FIN*-learn H_k from this text, contrary to assumption. Therefore $(\forall k)[H_k = H_i \vee t_m^+ \not\subseteq H_k]$ as required for the first condition in Definition 18. The second condition is easily checked.

f is an inner sufficiency function over $\mathcal{L}$: by assumption, any text t for any $H \in \mathcal{C}$ results in the output of some k with $H_k = H$ at some stage n when the

hypotheses in play are $H_1, ..., H_j$. Then by definition of f, we have $f(t_n, j) = 1$, as required.

f is recursive: to recursively compute $f(t_m, n)$, we run the Turing Machine $\mathcal{M}$ on t_m with the hypotheses $H_1, ..., H_n$ in play. If $\mathcal{M}$ outputs a hypothesis i on input t_m then (obviously $i \leq n$) set $f(t_m, n) = 1$. If $\mathcal{M}$ requests more input then set $f(t_m, n) = 0$.

□

Corollary 11 *There exists a canonical FIN-learner which FIN-learns any $\mathcal{C} \in$ FIN-TXT w.r.t. any $\mathcal{L} \in$ Index$(\mathcal{C})$ using any recursive inner sufficiency function over $\mathcal{L}$ as an oracle.*

Proof

The Turing Machine $\mathcal{M}$ constructed in the $\Leftarrow$ proof of Theorem 7 is such a learner as the definition of $\mathcal{M}$ does not depend on $\mathcal{C}$ except via f.

□

We may use a sufficiency function to define a particularly strong form of corroboration function.

Definition 20 *$c(H, t)$ is called a* sufficient corroboration function *over $\mathcal{L}$ if there exists an inner sufficiency function $f(t, n)$ over $\mathcal{L}$ such that:*

$$(\forall t)(\forall i)(\forall m)[[c(H_i, t_m) > 0 \wedge c(H_i, t_m) \in c'(H_i)] \Rightarrow f(t_m, i) = 1]$$

and

$$(\forall t)(\forall m)(\forall n)[f(t_m, n) = 1 \Rightarrow (\exists i \leq n)c(H_i, t_m) \in c'(H_i)]$$

where

$$c'(H_i) = \{c(H_i) \mid H_i = H \wedge (\forall j)[c(H_j) > c(H_i) \Rightarrow H_j \neq H]\}$$

c is called a recursive sufficient corroboration function *if both c and $c'_f : \mathcal{L} \times S \to \{0, 1\}$ defined by*

$$c'_f(H_i, s) = \begin{cases} 1 & \text{if } s \in c'(H_i) \\ 0 & \text{otherwise} \end{cases}$$

are total and recursive.

Theorem 8 *$\mathcal{C} \in$ FIN-TXT iff there exists $\mathcal{L} \in$ Index$(\mathcal{C})$ such that there exists a recursive sufficient corroboration function c over $\mathcal{L}$.*

Proof

($\Leftarrow$) We define a *recursive* inner sufficiency function f' over $\mathcal{L}$ based on c. The result then follows from Theorem 7.

Define

$$f'(t_m, n) = \begin{cases} 1 & \text{if } (\exists i \leq n) c(H_i, t_m) \in c'(H_i) \\ 0 & otherwise \end{cases}$$

f' is recursive: to compute $f'(t_m, n)$, only finitely many recursive computations of $c(H_i, t_m)$ and $c'_f(H_i, c(H_i, t_m))$ are needed to test the condition above.

f' is an inner sufficiency function: let f be the sufficiency function over $\mathcal{L}$ which we know to exist from Definition 20. Let t be a text for some $H_i \in \mathcal{L}$. There exist $n \geq i$ and m such that $f(t_m, n) = 1$, by Definition 19. Then $c(H_i, t_m) \in c'(H_i)$ by Definition 20 and finally $f'(t_m, n) = 1$ by our construction of f'.

($\Rightarrow$) Suppose $\mathcal{L} \in$ *FIN-TXT*. Let $\mathcal{M}$ be a learner which *FIN*-learns $\mathcal{C}$ w.r.t. $\mathcal{L}$ and f be the recursive inner sufficiency function from the ($\Rightarrow$) proof of Theorem 7. We define c using $\mathcal{M}$ as follows. Let t be any text.

$$c(H_i, t_m) = \begin{cases} -1 & \text{if } t_m \text{ refutes } H_i \\ 1 & \text{if } \mathcal{M}(t_m) = i \\ 0 & \text{otherwise} \end{cases}$$

c is recursive: obvious based on the recursiveness of $\mathcal{M}$.

c'_f is total and recursive: immediate since $(\forall i) c'(H_i) = \{1\}$.

c is a sufficient corroboration function over $\mathcal{L}$: it is easily checked that the conditions of Definition 20 are satisfied.

□

Corollary 12 *There exists a canonical FIN-learner with corroboration which FIN-learns any $\mathcal{C} \in$ FIN-TXT w.r.t. any $\mathcal{L} \in Index(\mathcal{C})$ using any recursive sufficient corroboration function over $\mathcal{L}$ as an oracle.*

Proof

Immediate from proof of Theorem 8 ($\Leftarrow$) as the definition of $\mathcal{M}$ does not depend on $\mathcal{C}$ except via c.

□

The existence of a recursive sufficient corroboration function over some $\mathcal{L} \in Index(\mathcal{C})$ for *FIN*-learning of $\mathcal{C}$ to succeed is a very strong requirement. This should not surprise us. *FIN*-learning is an identification criterion far removed from Popper's dictum that although a hypothesis may be very strongly corroborated, it is never (in normal circumstances) safe from later refutation. Only when the hypothesis space is unusual in some respect can such a corroboration function exist.

Theorem 9 *$\mathcal{C} \in$ JREF-TXT iff there exists $\mathcal{L} \in Index(\mathcal{C})$ such that there exists a recursive outer sufficiency function f over $\mathcal{L}$ and a recursive limiting corroboration function c over $\mathcal{L}$.*

Proof

($\Leftarrow$) Let $\mathcal{L} = H_1, H_2, ... \in Index(\mathcal{C})$ and let f and c be a recursive outer sufficiency function over $\mathcal{L}$ and a recursive limiting corroboration function over $\mathcal{L}$, respectively. We define our inductive *JREF*-learner $\mathcal{M}'$ based on the canonical learner $\mathcal{M}$ from the proof of Theorem 2($\Leftarrow$), as follows. Recall that $\perp$ is the special symbol output by $\mathcal{M}$ to refute the entire hypothesis space, prior to halting.

$$\mathcal{M}'(t_m) = \begin{cases} \perp & \text{if } f(t_m, p) = 1 \wedge (\forall i \leq p) t_m \text{ refutes } H_i \\ \mathcal{M}(t_m) & \text{otherwise} \end{cases}$$

where $H_1, ..., H_p$ are the hypotheses in play at stage m.

$\mathcal{M}'$ is recursive: follows immediately from the recursiveness of f and $\mathcal{M}$.

On presentation of a text t for $H \in \mathcal{C}$, $\mathcal{M}$ converges to some j such that $H_j = H$: in this case there are no n, m on which $f(t_m, n) = 1$ so the behaviour of $\mathcal{M}'$ is identical to that of $\mathcal{M}$ on the same text. The result follows from the $\Leftarrow$ proof of Theorem 2.

On presentation of a text $t \notin Txts(\mathcal{L})$, $\mathcal{M}'$ eventually outputs the symbol $\perp$ and halts: by assumption f is an outer sufficiency function so by Definition 19, there exist m, n such that $f(t_m, n) = 1$. Also there exists some m' such that $t_{m'}$ refutes all of $H_1, ..., H_n$, so we are done.

($\Rightarrow$)

Suppose $\mathcal{M}$ *JREF*-learns $\mathcal{C}$ w.r.t. $\mathcal{L}$ and let t be any text. Define f as follows:

$$f(t_m, n) = \begin{cases} 1 & \text{if } (\exists m' \leq m)[\mathcal{M}(t_{m'}) = \perp \\ & \wedge \text{ hypotheses in play at stage } m' \text{ are } H_1, ..., H_p \text{ where } p \leq n] \\ 0 & \text{otherwise} \end{cases}$$

f is recursive: immediate from recursiveness of $\mathcal{M}$.

f is an outer sufficiency function for $\mathcal{L}$: by assumption that $\mathcal{M}$ *JREF*-learns $\mathcal{C}$ w.r.t $\mathcal{L}$, on any text $t \notin Txts(\mathcal{L})$, $\mathcal{M}$ outputs the symbol $\perp$ and halts, say at stage m when hypotheses in play are $H_1, ..., H_p$. Then $f(t_m, p) = 1$, as required. The conditions of Definition 18 are trivially satisfied.

Finally, the recursive limiting corroboration function from the proof of Theorem 2($\Rightarrow$) has the desired properties for our c.

□

As in the discussion of *FIN-TXT*, the learning criterion *JREF-TXT* has a very un-Popperian aspect, and consequently the necessary and sufficient condition for $\mathcal{C} \in$ *JREF-TXT* is very strong.

We have not found it possible to give an analogous condition for the form of refutational learning (*REF-TXT*) in [MA93], as that identification criterion has a non-effective element to its definition.

7 Examples

The corroboration functions constructed in the $\Rightarrow$ proofs in Section 6 were simplistic. However in practical use, the existence or non-existence of appropriate corroboration functions may be suggested naturally by the space of hypotheses in use. We give some examples of the use of corroboration functions to prove the learnability or otherwise under certain identification criteria of some simple examples.

Our example languages will be sets of points in the rational plane $\mathcal{Q}^2$, so $\Sigma = \{(a,b) \mid a,b \in \mathcal{Q}\}$.

Example 1 *Let $\mathcal{C}$ be the set of all closed circles of finite radius. Let $<,>$ be a fixed recursive bijection between $\mathcal{Q}^2$ and $\mathbb{N}$ and $<<,>>$ a fixed recursive bijection between $\mathcal{Q}^2$ and $\mathcal{Q}$. A suitable hypothesis space $\mathcal{L} = H_1, H_2, \ldots$ is given by*

$$H_{<a,b>} = \{(p,q) \mid a = << x,y >> \wedge (p-x)^2 + (q-y)^2 \leq b^2\}$$

It is easily seen that $\mathcal{L}$ is an indexing of $\mathcal{C}$.

Consider the following corroboration function $c : \mathcal{L} \times (\Sigma^*) \rightarrow \mathcal{Q} \cup \{\infty\}$, which is based on the naturalistic idea that the further away a point is from a, the more severe a test it is of hypothesis $H_{<a,b>}$. For circles of non-zero radius b we also include a scaling multiplier of $1/b^2$ into the corroboration function, so that smaller circles are potentially more highly corroborable than large ones.

$$c(H_{<a,b>}, t_m) = \begin{cases} 0 & \textit{if } t_M^+ = \emptyset \\ -1 & \textit{if } t_m \textit{ refutes } H_{<a,b>}, \\ & \quad i.e.\ [a = << x,y >> \\ & \quad \wedge (\exists (c,d) \in t_m^+)[(c-x)^2 + (d-y)^2 > b^2]] \\ \infty & \textit{if } b = 0 \wedge a = << x,y >> \wedge t_m^+ = \{(x,y)\} \\ 1/b^2 * max(a,b,t_m) & \textit{otherwise} \end{cases}$$

where

$$max(a,b,t_m) = max\{((c-x)^2 + (d-y)^2)/b^2 \mid a = << x,y >> \wedge (c,d) \in t_m^+\}$$

With a little checking we see that c is indeed a corroboration function under Definition 11, and is recursive and natural. c is limiting because on any text t for H_i we have a stage m at which t_m contains two diametrically opposed points on the circumference of the circle defined by H_i. Then if we let $i = < a,b >$:

- $(\forall j)[[j =< c, d > \wedge d^2 < b^2] \rightarrow [t_m \textit{ refutes } H_j \wedge c(H_j, t_m) = -1]]$.
- $(\forall j)[[j =< c, d > \wedge d^2 > b^2] \rightarrow (\forall n \geq m)c(H_i, t_n) = 1/b^2 > 1/d^2 \geq c(H_j, t_n)$
- $(\forall j)[[j =< c, d > \wedge d^2 = b^2] \rightarrow [c = a \vee t_m \textit{ refutes } H_j]]$

These are the only cases, so at all stages $n \geq m$ we have that $H_{<a,b>}$ is the most strongly corroborated hypothesis (except for $H_{<a,-b>}$, which is equally strongly corroborated and describes the same circle).

c is also attaining because

- *if* $b = 0$ *then* $(\forall a)c(H_{<a,0>}) = \infty$
- *if* $b \neq 0$ *then* $(\forall a)c(H_{<a,b>}) = 1/b^2$

and for example $c(H_{<a,b>}, t_0) = c(H_{<a,b>})$ where $t = (x + b, y), ...$ is a text for $H_{<a,b>}$ and $a =<< x, y >>$.

The above suffices to prove that $\mathcal{C} \in$ s-CONSERV-TXT, by Theorem 5.

Finally we can see that c is not strict because for example (let $b > 0$) $t = (x+b, y), ...$ results in $c(H_{<<<x,y>>,b>}, t_0) = 1/b^2 = c(H_{<<<x,y>>,b>})$ although many hypotheses H_j with $H_{<<<x,y>>,b>} \not\subseteq H_j$ remain unrefuted. Nevertheless it is possible to find a recursive, strict, attaining, limiting, set-driven corroboration function over $\mathcal{L}$ by requiring that two diametrically opposed points on the circumference of H_i must appear in the text before we set $c(H_i, t_m) = c(H_i)$. This proves that $\mathcal{C} \in$ s-SMON-TXT. The details are left as an exercise for the reader.

Example 2 *Now let $\mathcal{C}$ be the set of all* open *circles of finite radius. We show that $\mathcal{C} \notin$ LIM-TXT as follows.*

Let $\mathcal{L}$ be an indexing of $\mathcal{C}$ and suppose for a contradiction that c is a recursive limiting corroboration function over $\mathcal{L}$. Let $\mathcal{M}$ be the canonical LIM-learner from Corollary 4.

Let H_i be a hypothesis for the circle centre a, radius $b > 0$. We construct a text t for H_i such that $\mathcal{M}$ fails to converge to an index for H_i using oracle c.

Begin enumerating H_i as the text t. Suppose after m stages we have $\mathcal{M}(t_m) = j$ where $H_j = H_i$. Now however there exists a circle H_k with centre a, radius $b - \varepsilon$ for some suitably small ε, such that $t_m^+ \subset H_k$. Continue t by enumerating further points in H_k.

There are two cases. If $(\forall n \geq m)Best_n = j$, then t is a text for H_k on which $\mathcal{M}$ fails to converge to an index for H_k. Otherwise at some stage $n > m$, $Best_n \neq j$. In this case t resumes enumerating H_i. This construction can be repeated infinitely often, so $Best_m$ fails to converge on t, a text for H_i. In either case $\mathcal{M}$ fails to LIM-learn $\mathcal{C}$ w.r.t. $\mathcal{L}$, a contradiction by Corollary 4. We conclude that $\mathcal{C} \notin$ LIM-TXT.

8 Artificial Intelligence and Induction

We will briefly discuss a recent body of work by Gillies [Gi93, Gi96] which puts a 'Baconian' interpretation on certain successful developments in machine learning.

Gillies contends, *contra* Popper's belief that the creation of scientific theories is not mechanisable, nor even amenable to logical (as opposed to psychological) study, that modern machine learning algorithms behave in a highly Baconian manner; this he describes as *mechanical falsification.* In short, such learners synthesise hypotheses from background knowledge and existing evidence before subjecting them to the risk of Popperian refutation by later evidence. This rolls back the creative element of discovery to the higher level problem of deciding which is the appropriate background knowledge to use - in our parlance, which is the appropriate hypothesis space. He posits, furthermore, that this is the first time in the history of science that Bacon's inductivism has really been used, for prior to machine learning no general method was given to enable the learner/discoverer to mechanically (ie. without intelligence) produce hypotheses - one of Bacon's stated aims.

It is difficult to deny that machine learners (e.g. ID3 [Qu79], GOLEM [MF92]) do indeed behave in this way. Gillies's theme throughout [Gi96] is that logic has both inferential and control components - in learning or discovery the inference corresponds to Popperian falsification from data, while the control element lies in the production of new hypotheses. Gillies specifically mentions degree of corroboration as just such a control element.

Our work in this paper is entirely in accord with Gillies's view, particularly our use of corroboration functions as a control element (indeed given the characterisation results with canonical learners which we have obtained, as the sole control element) in inductive learning.

The results obtained in the case of refuting learning are particularly interesting viewed from this angle. Classic examples from the history of science such as Kepler's laws of planetary motion (see again [Gi96]) demonstrate that it is not the synthesising of a hypothesis from data and background knowledge that constitutes great science, but the paradigm shift that results from a change in background knowledge or assumptions. This corresponds to the various forms of refuting inductive inference which have been defined (Section 6.4) and suggests that the truly creative machine learner will not only be able to learn within or refute an existing hypothesis space, but also to propose a new one. Such a learner is unlikely to be developed soon.

9 Conclusions and Future Work

We have proposed a unifying model for machine inductive inference based on the philosophical work of K.R. Popper, and obtained characterisations of many of the standard identification types in learning indexed families of recursive languages from text. In our model canonical learners use recursive oracles which compute a version of Popper's degree of corroboration. These learners then follow the natural strategy of preferring the most strongly (or at least a maximally strongly) corroborated hypothesis at any given time. Membership of a class of concepts within a particular identification criterion is then equivalent to the existence of a recursive corroboration function with certain properties depending on the identification type.

We intend to extend this unifying model of learning to include language learning from informant and related problems such as learning of recursive functions. An extension of our approach to learning from noisy data would be particularly interesting; in this case it is no longer certain that a single adverse data item refutes a hypothesis and we would be obliged to allow negative corroboration values other than -1, as in Popper's original model. Given the crucial role played by the hypothesis space in our model, it would also be interesting to extend this approach to cover exact and class comprising learning.

Acknowledgements

The author wishes to thank Prof. Dr. Steffen Lange of Universität Leipzig for his comments on an earlier draft, which significantly improved Sections 6.1 and 6.2.

References

[An80] D. Angluin, Inductive inference of formal languages from positive data, *Information and Control* 45, 117-135, 1980.
[CL82] J. Case, C. Lynes, Machine inductive inference and language identification, in M. Nielsen, E.M. Schmidt (Eds.), Proc. of the Ninth International Colloquium on Automata, Languages and Programming, Springer LNCS 140, 107-115, 1982.
[Fu90] M. Fulk, Prudence and other restrictions in formal language learning, *Information and Computation* 85, 1-11, 1990.
[Gi93] D. Gillies, *Philosophy of Science in the Twentieth Century*, Blackwell, 1993.
[Gi96] D. Gillies, *Artificial Intelligence and Scientific Method*, Oxford University Press, 1996.
[Go67] E.M. Gold, Language identification in the limit, *Information and Control* 10, 447-474, 1967.

[Ja91] K.P. Jantke, Monotonic and non-monotonic inductive inference, *New Generation Computing* 8, 349-460.
[LW94] S. Lange P. Watson, Machine discovery in the presence of incomplete or ambiguous data, in S. Arikawa, K.P. Jantke (Eds.) Algorithmic Learning Theory, Proc. of the Fifth International Workshop on Algorithmic Learning Theory, Rheinhardsbrunn, Germany, Springer LNAI 872, 438-452, 1994.
[LZ93] S. Lange, T. Zeugmann, Monotonic versus non-monotonic language learning, in G. Brewka, K.P. Jantke, P.H. Schmitt (Eds.), Proc. of the Second International Workshop on Nonmonotonic and Inductive Logic, Springer LNAI 659, 254-269, 1993.
[LZ94] S. Lange, T. Zeugmann, Set-driven and rearrangement-independent learning of recursive languages, in S. Arikawa, K.P. Jantke (eds.) Algorithmic Learning Theory, Proc. of the Fifth International Workshop on Algorithmic Learning Theory, Rheinhardsbrunn, Germany, Springer LNAI 872, 453-468, 1994.
[MA93] Y. Mukouchi, S. Arikawa, Inductive inference machines that can refute hypothesis spaces, in K.P. Jantke, S. Kobayashi, E. Tomita, T. Yokomori (Eds.), Algorithmic Learning Theory, Proc. of the Fourth International Workshop on Algorithmic Learning Theory, Tokyo, Japan, Springer LNAI 744, 123-136, 1993.
[MF92] S. Muggleton, C. Feng, Efficient induction of logic programs, in S. Muggleton (Ed.) *Inductive Logic Programming*, Academic Press, Ch. 13, 281-298, 1992.
[OW82] D. Osherson, S. Weinstein, Criteria of language learning, *Information and Control* 52, 123-138, 1982.
[Po34] K.R. Popper, *The Logic of Scientific Discovery*, 1997 Routledge reprint of the 1959 Hutchinson translation of the German original.
[Po54] K.R. Popper, Degree of confirmation, *British Journal for the Philosophy of Science* 5, 143ff, 334, 359, 1954.
[Po57] K.R. Popper, A second note on degree of confirmation, *British Journal for the Philosophy of Science* 7 350ff, 1957.
[Po63] K.R. Popper, *Conjectures and Refutations*, Routledge, 1963 (Fifth Edition, 1989).
[Qu79] J.R. Quinlan, Discovering rules by induction from large collections of examples, in D. Michie (Ed.), *Expert Systems in the Micro-Electronic Age*, Edinburgh University Press, 168-201, 1979.
[Sc84] G. Schäfer-Richter, *Uber Eingabeabhängigkeit und Komplexität von Inferenzstrategien*, Dissertation, Rheinisch Westfälische Technische Hochschule Aachen, 1984.
[WC80] K. Wexler, P. Culicover, *Formal Principles of Language Acquisition*, MIT Press, Cambridge, MA, 1980.
[Wi91] R. Wiehagen, A thesis in inductive inference, in J. Dix, K.P. Jantke and P.H. Schmitt (Eds.), Proc. of the First International Workshop on Nonmonotonic and Inductive Logic, Karlsruhe, 1990, Springer LNAI 543, 184-207, 1991.