T H E N A T I O N A L U N I V E R S I T Y
o f S I N G A P O R E

S c h o o l o f C o m p u t i n g
Computing 1, 13 Computing Drive, Singapore 117417

TRB4/09

Iterative Learning from Texts and Counterexamples Using Additional Information

Sanjay Jain and Efim Kinber

April 2009

Technical Report

Foreword

This technical report contains a research paper, development or tutorial article, which has been submitted for publication in a journal or for consideration by the commissioning organization. The report represents the ideas of its author, and should not be taken as the official views of the School or the University. Any discussion of the content of the report should be sent to the author, at the address shown on the cover.

OOI Beng Chin
Dean of School

Iterative Learning from Texts and Counterexamples Using Additional Information

Sanjay Jain*[1] and Efim Kinber[2]

[1] School of Computing, National University of Singapore, Singapore 117417, Republic of Singapore. Email: sanjay@comp.nus.edu.sg

[2] Department of Computer Science, Sacred Heart University, Fairfield, CT 06825-1000, U.S.A. Email: kinbere@sacredheart.edu

Abstract. A variant of *iterative* learning in the limit (cf. [LZ96]) is studied when a learner gets negative examples refuting conjectures containing data in excess of the target language and uses additional information of the following four types: a) memorizing up to n input elements seen so far; b) up to n *feedback* memberships queries (testing if an item is a member of the input seen so far); c) the number of input elements seen so far; d) the maximal element of the input seen so far. We explore how additional information available to such learners (defined and studied in [JK07]) may help. In particular, we show that adding the maximal element or the number of elements seen so far helps such learners to infer any *indexed* class of languages *class-preservingly* (using a descriptive numbering defining the class) — as it is proved in [JK07], this is not possible without using additional information. We also study how, in the given context, different types of additional information fare agains each other, and establish hierarchies of learners memorizing $n+1$ versus n input elements seen and $n+1$ versus n feedback membership queries.

1 Introduction

In this paper, we study some variants of learning in the limit from positive data and negative counterexamples to conjectures, with restricted access to input data. The general framework for study of learning in the limit was introduced in [Gol67]. In the Gold's original model, **TxtEx**, a learner is able to hold full input data seen so far in its long-term memory. However, this assumption is apparently too strong for modeling many learning and cognitive processes. Wiehagen in [Wie76] (see also [LZ96]) suggested a model for learning in the limit where the long-term memory of the learners is limited to what they can store in their conjectures. These learners are called *iterative* learners. This learning model, while strongly limiting long-term memory, still makes salient an important aspect of learnability in the limit: its incremental character. Some variants of iterative learning proved to be quite useful in the context of applied machine learning (for example, [LZ06] applies the idea of iterative learning in the context of training Support Vector Machines).

* Supported in part by NUS grant number R252-000-308-112.

Iterative learning model has been used for study of learnability from all positive examples (the corresponding formal model being denoted as **TxtIt**) as well as all positive and negative examples (denoted as **InfIt**, see [LZ92]). One can argue that **TxtIt** may be too weak (a learner gets only positive data and can memorize only very limited amount of input), whereas **InfIt** may be too strong: it is hard to conceive a realistic learning process, where the learner would be able to get access to full negative data. For example, children learning languages, while getting some negative data (in form of corrections by parents or teachers), never get the full set of negative data.

In [JK08], the model **TxtEx** was extended to allow *negative counterexamples* to conjectures by a learner. This model is an example of *active learning*, where a learner communicates with a *teacher* (formally, an oracle) making *queries* and getting responses from the teacher. Active learning as a general framework for study of learning processes was introduced by D. Angluin in [Ang88] and has been widely utilized in various studies of theoretical and applied models of learnability from examples since then. The model of iterative learning from full positive data and negative counterexamples, **NCIt** (NC here stands for *"negative counterexample"*), defined in [JK07] actually combines two approaches: the Gold's framework (as the learner incrementally gets access to full positive data) and active learning (the learner, using *subset* queries, checks with the teacher if each conjecture does not contain data in excess of the target languages — in particular, if the conjecture does not overgeneralize (it should be noted that K. Popper [Pop68] regarded refutation of overgeneralizing conjectures as a vital part of learning and discovery processes) — and if the answer is negative, the learner gets a *negative counterexample* showing an error (in linguistic terms, non-grammatical sentences in conjectures are, thus, being corrected).

In this paper, we extend **NCIt** model to incorporate some additional features. Specifically, we consider the following two extensions of this model: in addition to subset queries (for conjectures), the learner

a) can ask up to n *feedback* queries about the elements of input seen so far (and not saved);

b) can store up to n input elements seen so far in its long-term memory (note that when long-term memory used by a learner is n-bounded, in order to save a new input datum, the learner must sacrifice one element currently stored in the memory);

In the context of iterative learning languages from positive data, these two types of "looking back" (in the context of feedback — using just one query per conjecture) were defined in [LZ96] (an earlier variant of memory-bounded learning can be found in [OSW86], and the idea of feedback learning goes back to [Wie76], where it was applied in the context of learning recursive functions in the limit). Both these concepts were reformalized (the former named *n-feedback* learning, and the latter named *n-bounded memory* learning) and thoroughly studied and discussed in [CJLZ99]. Motivation for these sorts of learnability models, as discussed in [CJLZ99], comes from the rapidly developing field of knowledge discovery in databases, which includes, in particular, data mining, knowledge extraction, information discovery, data pattern processing, information harvesting, etc. Many of these tasks represent interactive incremental iterative processes (cf., e.g., [BA96] and [FPSS96]), working on huge data sets, finding regularities, and verifying them on small samples of the overall data. While the authors in [CJLZ99] explore the abovementioned formalizations of “looking back” at small (uniformly limited by some upper bound n) portions of input data in the context of regular iterative learning, we, in this research, allow the learner to test with the teacher if conjectures do not contain data in excess of the target language. Our learners may also be allowed to memorize some “bounds” derived from the input data seen so far — in form of the maximal element or the length of input seen so far. In this research, we study how the abovementioned types of additional information can enhance capabilities of **NCIt** learnability model in general, and how they, while helping a learner, fare against each other.

Specifically, in Section 3, we discover some general effects of additional information on **NCIt**-learners. In particular, it was established in [JK07] that iterative learners getting access to full positive and full negative data are, surprisingly, weaker than **NCIt**-learners (note that the latter ones get negative data just in form a finite set of negative counterexamples — however, only when these negative data is really necessary). We now show that when learners getting full positive and full negative data are allowed to memorize just one input datum or ask just one feedback membership query, they can sometimes learn more than any learner that gets access to full positive data and can use negative counterexamples (to conjectures) and store *all* data seen so far in its long-term memory (see Theorem 9). A known capability of **NCIt**-learners (established in [JK07]) is of special importance for many practical classes of languages: they can learn every indexed class of languages (that is, any class of recursive languages, where it is decidable, for any index k and any element m, if m is a member of the language with index k; examples of such classes are the class of all regular languages and all *patterns*). However, as it was established in [JK07], **NCIt**-learners sometimes cannot learn an indexed class *class-preservingly* (cf. [ZL95]) — that is, they cannot learn by using any descriptive numbering defining just the target class as hypothesis space. It turns out that this feature of **NCIt**-learners holds even if they can make n-feedback membership queries (see Theorem 12). However, class-preserving learning becomes possible if an **NCIt**-learner gets access to either the maximal element or to the number of elements seen so far (see Theorem 11).

In Section 4, we strengthen some results in [CJLZ99], establishing non-trivial hierarchies of **NCIt**-learners using n-feedback queries or n-bounded memory based on the number n (see Theorems 13 and 14). Our examples of classes witnessing the hierarchies in question also show that additional information in form of the maximal element seen so far and the number of elements seen so far might not match the help that an **NCIt**-learner gets in form of one extra feedback membership query, or one extra long-term memory cell.

In Section 5, we study tradeoffs between different types of additional information used by **NCIt**-learners (the main purpose of this study is to make salient advantages of each type of additional information for the learners in question). In particular, similarly to corresponding results in [CJLZ99], we show that one memory cell used by an **NCIt**-learner can give more help than any n feedback membership queries (even in presence of the maximal element and the number of elements seen so far), see Theorem 15, and, conversely, one feedback membership query can give more help than n-bounded memory (plus the maximal element and the number of elements seen so far), see Theorem 16. Interestingly, the maximal element seen so far alone can give more help than any number of feedback membership queries (Theorem 20). Also, the number of elements and the maximal element seen so far combined together can provide more help than any bounded number of memory cells or feedback membership queries, Theorem 22. We also show how an extra memory cell can simulate maximal element for **NCIt**-learners using n memory cells, Proposition 17. We also obtain some partial results for other possible tradeoffs. Additionally, in Section 6 we also demonstrate that yet another type of additional information, the length of input seen so far (note that this number may be, apparently, different from the number of elements in the input seen so far), in the context of **NCIt**-learning, can be replaced by the maximal element seen so far, Theorem 29.

In Section 7, we briefly address the issue of robustness of our results using the maximal element and/or the number of elements seen so far in the presence of some "noise". Namely, we discuss which results hold if the maximal element and the number of elements seen so far are replaced by upper bounds on these numbers, or by some approximations of these numbers.

2 Preliminaries

2.1 Notation

For any unexplained recursion theoretic notation we refer the reader to [Rog67]. The symbol $\mathbb{N}$ denotes the set of natural numbers, $\{0, 1, 2, 3, \ldots\}$. Languages are subsets of $\mathbb{N}$. Symbols $\emptyset$, $\subseteq$, $\subset$, $\supseteq$, and $\supset$ respectively denote empty set, subset, proper subset, superset, and proper superset. Cardinality of a set S is denoted by $card(S)$. The maximum and minimum of a set are denoted by $max(\cdot), min(\cdot)$, respectively, where $max(\emptyset) = 0$ and $min(\emptyset) = \infty$. $\overset{\infty}{\forall}$ denotes 'for all but finitely many'.

We let D_x denote the finite set with canonical index x [Rog67]. We let $\langle \cdot, \cdot \rangle$ stand for an arbitrary, computable, 1–1 mapping from $\mathbb{N} \times \mathbb{N}$ onto $\mathbb{N}$ [Rog67]. We assume without loss of generality that $\langle \cdot, \cdot \rangle$ is monotonically increasing in both of its arguments. We define $\pi_1^2(\langle x, y \rangle) = x$ and $\pi_2^2(\langle x, y \rangle) = y$. Pairing function can be extended to n-tuples in a natural way (for example, by using $\langle x, y, z \rangle = \langle x, \langle y, z \rangle \rangle$). The corresponding projection functions are $\pi_i^n(x_1, x_2, \ldots, x_n) = x_i$.

By φ we denote a fixed *acceptable* programming system for the partial computable functions from $\mathbb{N}$ to $\mathbb{N}$ [Rog67,HU79]. By φ_i we denote the partial computable function computed by the program with number i in the φ-system. For a partial function η, $\eta(x)\downarrow$ denotes that $\eta(x)$ is defined. $\eta(x)\uparrow$ denotes that $\eta(x)$ is undefined.

By Φ we denote an arbitrary fixed Blum complexity measure [Blu67,HU79] for the φ-system. Intuitively, $\Phi_i(x)$ may be thought as the number of steps it takes to compute $\varphi_i(x)$.

By W_i we denote $\text{dom}(\varphi_i)$. Thus, W_i can be viewed as the recursively enumerable (r.e.) set/language accepted by the φ-program i. We also say that i is a grammar for W_i. By $W_{i,s}$ we denote the set $\{x < s : \Phi_i(x) < s\}$. $\mathcal{E}$ denotes the set of all r.e. languages. L, with or without decorations, ranges over $\mathcal{E}$. $\mathcal{L}$, with or without decorations, ranges over subsets of $\mathcal{E}$. χ_L denotes the characteristic function of L, and $\overline{L} = \mathbb{N} - L$, that is the complement of L.

$\text{pad}(j, \cdot, \cdot, \ldots)$ denotes a 1–1 recursive padding function (of appropriate number of arguments) such that $W_{\text{pad}(j,\cdot,\cdot,\ldots)} = W_j$ (such computable functions exist [Rog67]).

$\mathcal{L}$ is said to be an *indexed family* iff there exists an indexing $L_0, L_1, \ldots$ of all and only the languages in $\mathcal{L}$ such that for some recursive function f, $f(i,x) = \chi_{L_i}(x)$.

2.2 Basic Definitions for Learning

A *text* T is a mapping from $\mathbb{N}$ into $(\mathbb{N} \cup \{\#\})$. $T(i)$ represents the $(i+1)$-th element in the text. We let T, with or without decorations, range over texts. content(T) denotes the set of natural numbers in the range of T. A text T is for a language L iff content$(T) = L$. Intuitively, $T(i)$ denotes the element presented to the learner at time i, and #'s represent pauses in the presentation of data. $T[n]$ denotes the initial sequence of T of length n, that is $T[n] = T(0)T(1)\ldots T(n-1)$.

Sets of form $\{x : x < n\}$, for some n, are called initial segments of $\mathbb{N}$. A *(finite) sequence* σ is a mapping from an initial segment of $\mathbb{N}$ into $(\mathbb{N} \cup \{\#\})$. The empty sequence is denoted by λ. SEQ denotes the set of all finite sequences. We let σ, τ, and γ, with or without decorations, range over finite sequences. The *length* of σ, denoted by $|\sigma|$, is the number of elements in σ. For $n \leq |\sigma|$, $\sigma[n]$ denotes the initial sequence of σ of length n. Thus, $\sigma[0]$ is λ. content(σ) denotes the set of natural numbers in the range of σ. We denote the sequence formed by the concatenation of τ at the end of σ by $\sigma \diamond \tau$. For simplicity of notation, sometimes we omit $\diamond$, when it is clear that concatenation is meant.

An *informant* [Gol67] I is a mapping from $\mathbb{N}$ to $(\mathbb{N} \times \{0,1\}) \cup \{\#\}$ such that for no $x \in \mathbb{N}$, both $(x,0)$ and $(x,1)$ are in the range of I. content(I) = set of pairs in the range of I. We say that a I is an *informant* for L iff content$(I) = \{(x, \chi_L(x)) : x \in \mathbb{N}\}$. The *canonical informant* for L is the informant $(0, \chi_L(0))(1, \chi_L(1)) \ldots$. Intuitively, informants give both all positive and all negative data for the language being learned. $I[n]$ is the first n elements of the informant I.

An *inductive inference machine* (IIM) [Gol67] learning from texts is an algorithmic device which computes a (possibly partial) mapping from SEQ into $\mathbb{N}$. One can similarly define learners from informants and other modes of input as considered below. We use the term learner or learning machine as synonyms for inductive inference machines. We let M, with or without decorations, range over IIMs. $M(T[n])$ (or $M(I[n])$) is interpreted as the grammar (index for an accepting program) conjectured by the IIM M on the initial sequence $T[n]$ (or $I[n]$). We say that M converges on T to i, (written: $M(T)\!\downarrow = i$) iff $(\stackrel{\infty}{\forall} n)[M(T[n]) = i]$. Convergence on informants is similarly defined.

There are several criteria for an IIM to be successful on a language. In this paper we will be mainly concerned with explanatory (abbreviated **Ex**) criteria of learning.

Definition 1. [Gol67,CL82]

(a) M **TxtEx**-*identifies an r.e. language* L (written: $L \in \mathbf{TxtEx}(M)$) just in case for all texts T for L, $M(T[n])$ is defined for all n and $(\exists i : W_i = L)(\stackrel{\infty}{\forall} n)[M(T[n]) = i]$.

(b) M **TxtEx**-*identifies a class* $\mathcal{L}$ *of r.e. languages* (written: $\mathcal{L} \subseteq \mathbf{TxtEx}(M)$) just in case M **TxtEx**-identifies each language from $\mathcal{L}$.

(c) $\mathbf{TxtEx} = \{\mathcal{L} \subseteq \mathcal{E} : (\exists M)[\mathcal{L} \subseteq \mathbf{TxtEx}(M)]\}$.

Definition 2. [Gol67,CL82] (a) M **InfEx**-*identifies an r.e. language* L (written: $L \in \mathbf{InfEx}(L)$), just in case for all informants I for L, $M(I[n])$ is defined for all n and $(\exists i : W_i = L)(\stackrel{\infty}{\forall} n)[M(I[n]) = i]$.

(b) M **InfEx**-*identifies a class* $\mathcal{L}$ *of r.e. languages* (written: $\mathcal{L} \subseteq \mathbf{InfEx}(M)$) just in case M **InfEx**-identifies each language from $\mathcal{L}$.

(c) $\mathbf{InfEx} = \{\mathcal{L} \subseteq \mathcal{E} : (\exists M)[\mathcal{L} \subseteq \mathbf{InfEx}(M)]\}$.

Next we consider iterative learning.

Definition 3. [Wie76,LZ96]

(a) M is iterative, iff there exists a partial recursive function F such that, for all T and n, $M(T[n+1]) = F(M(T[n]), T(n))$. Here $M(\lambda)$ is viewed as some predefined hypothesis.

(b) M **TxtIt**-*identifies* $\mathcal{L}$, iff M is iterative, and M **TxtEx**-identifies $\mathcal{L}$.

(c) $\mathbf{TxtIt} = \{\mathcal{L} : (\exists M)[M \ \mathbf{TxtIt}\text{-identifies } \mathcal{L}]\}$.

InfIt can be defined similarly. Note that for **InfIt**-learning, the learner has to succeed on all informants, and not only on the canonical one.

Intuitively, an iterative learner [Wie76,LZ96] is a learner whose hypothesis depends only on its last conjecture and current input. That is, for some recursive function F, for $n \geq 0$, $M(T[n+1]) = F(M(T[n]), T(n))$. Here, note that $M(T[0])$ is predefined to be some constant value. We will often identify F above with M (that is use $M(p, x) = F(p, x)$ to describe $M(T[n+1])$, where $p = M(T[n])$ and $x = T(n)$). This is for ease of notation. Context determines which interpretation of the learner M is meant.

For **Ex** models of learning (for learning from texts or informants or their variants when learning from positive data and negative counterexamples, as defined below), one may assume without loss of generality that the learners are total, that is, defined on all initial segments of all texts (see, for example [OSW86]). However for iterative learning one cannot assume so. Thus, we explicitly require in the definition that iterative learners are defined on all inputs which are initial segments of texts (informants) for a language in the class.

Note that, although it is not stated explicitly, an **It**-type learner might store some input data in its conjecture (thus serving as a limited long-term memory). However, the amount of stored data cannot grow indefinitely, as the learner must stabilize to one (right) conjecture.

We let $M_0, M_1, \ldots$ denote a recursive enumeration of all (iterative) IIMs from texts/informants or negative counterexamples, based on context.

Definition 4. (a) [Ful90] σ is said to be a **TxtEx**-*stabilizing sequence* for M on L, iff (i) $\mathrm{content}(\sigma) \subseteq L$, and (ii) for all τ such that $\mathrm{content}(\tau) \subseteq L$, $M(\sigma\tau) = M(\sigma)$.

(b) [BB75,Ful90] σ is said to be a **TxtEx**-*locking sequence* for M on L, iff (i) σ is a **TxtEx**-stabilizing sequence for M on L and (ii) $W_{M(\sigma)} = L$.

If M **TxtEx**-identifies L, then every **TxtEx**-stabilizing sequence for M on L is a **TxtEx**-locking sequence for M on L. Furthermore, one can show that if M **TxtEx**-identifies L, then for every σ such that $\mathrm{content}(\sigma) \subseteq L$, there exists a **TxtEx**-locking sequence, which extends σ, for M on L (see [BB75,Ful90]).

Similar results can be shown for **InfEx**, **TxtIt**, **InfIt** and other criteria of learning discussed in this paper. We will often drop **TxtEx** (and other criteria notation) from **TxtEx**-stabilizing sequence and **TxtEx**-locking sequence, when the criterion is clear from context.

Learning with feedback and learning with bounded memory is a generalization of iterative learning where the learner has access to some past data using queries or via some finite amount of memory. Thus, in feedback learning an iterative learner is additionally allowed to query whether some elements were present in the past input data. In bounded memory, an iterative learner is able to memorize in its memory some (bounded) finite number of data (in addition to its latest conjecture). Below are the formal definitions.

Definition 5. [CJLZ99] (a) Suppose M is a learning machine (for a class $\mathcal{L}$ of languages). We say that M is an m-feedback learner iff there exist partial recursive functions F and Q such that for all $L \in \mathcal{L}$, and all texts T for L,

(i) for all n: $Q(M(T[n]), T(n))\downarrow \in \mathbb{N}^m$, and

(ii) If $(Q(M(T[n]), x) = (x_1, x_2, \ldots, x_m)$ then $M(T[n{+}1]) = F(M(T[n]), T(n), y_1, y_2, \ldots, y_m)$, where $y_i = 1$ iff $x_i \in \text{content}(T[n])$.

(b) We say that M **TxtIt**-identifies $\mathcal{L}$ with m-feedback iff M **TxtEx**-identifies $\mathcal{L}$ and M is a m-feedback learner. Such learners M are also called **TxtIt**-learners using m-feedback.

Definition 6. [LZ96] (a) Suppose M is a learning machine (for a class $\mathcal{L}$ of languages). We say that M is an m-memory-bounded learner iff there exists a (partial) recursive memory function **mem** (mapping finite sequences to finite sets) and partial recursive functions F, F' such that for all $L \in \mathcal{L}$, and all texts T for L,

(i) for all n: $\mathbf{mem}(T[n])\downarrow \subseteq \text{content}(T[n])$ and $card(\mathbf{mem}(T[n])) \leq m$

(ii) for all n: $\mathbf{mem}(T[n+1]) = F'(M(T[n]), T(n), \mathbf{mem}(T[n]))\downarrow$, and $\mathbf{mem}(T[n+1]) - \mathbf{mem}(T[n]) \subseteq \{T(n)\}$.

(iii) $M(T[n+1]) = F(M(T[n]), T(n), \mathbf{mem}(T[n]))\downarrow$.

(b) We say that M **TxtIt**-identifies $\mathcal{L}$ with m-memory iff M **TxtEx**-identifies $\mathcal{L}$ and M is a m-memory-bounded learner. Such learners M are also called **TxtIt**-learner using m-memory or m-memory bounded **TxtIt**-learner.

In both the above definitions, $M(T[0])$ is some fixed initial hypothesis.

Again, we often identify the learner M with the function F (along with identifying **mem** with F') as defined above, and the context determines which interpretation of the learner M is meant.

One can similarly define feedback and memory bounded learning for learning from informants.

Besides above models of learning, we sometimes allow the learner access to the maximal element in the input seen so far, or the number of elements in the input seen so far as an additional input besides the input element $T(n)$ and the latest conjecture and feedback/memory (and the counterexamples, in the case of **NC**-type learning defined below). In the sequel, we will typically refer to the "maximal element seen so far" and the "number of elements seen so far" as simply the "maximal element" and, respectively, the "number of elements".

2.3 Learning with Negative Counterexamples

In this section we consider our models of learning from full positive data and negative counterexamples as given by [JK08]. Intuitively, for learning with negative counterexamples, we may consider the learner being provided a text, one element at a time, along with a negative counterexample to the latest conjecture, if any. (One may view this negative counterexample as a response of the teacher to the *subset query* when it is tested if the language generated by the conjecture is a subset of the target language). One may model the list of negative counterexamples as a second text for negative counterexamples being provided to the learner. Thus the IIMs get as input two texts, one for positive data, and other for negative counterexamples.

We say that $M(T, T')$ converges to a grammar i, iff $(\overset{\infty}{\forall} n)[M(T[n], T'[n]) = i]$.

First, we define the model of learning from positive data and negative counterexamples. In this model, if a conjecture contains elements not in the target language, then a negative counterexample is provided to the learner. **NC** in the definition below stands for *negative counterexample.*

Definition 7. [JK08]

(a) M **NCEx**-*identifies a language* L (written: $L \in \mathbf{NCEx}(M)$) iff for all texts T for L, and for all T' satisfying the condition:

$$(T'(n) \in S_n, \text{ if } S_n \neq \emptyset) \text{ and } (T'(n) = \#, \text{ if } S_n = \emptyset),$$
$$\text{where } S_n = \overline{L} \cap W_{M(T[n],T'[n])}$$

$M(T, T')$ converges to a grammar i such that $W_i = L$.

(b) M **NCEx**-*identifies a class* $\mathcal{L}$ *of languages* (written: $\mathcal{L} \subseteq \mathbf{NCEx}(M)$), iff M **NCEx**-identifies each language in the class.

(c) $\mathbf{NCEx} = \{\mathcal{L} : (\exists M)[\mathcal{L} \subseteq \mathbf{NCEx}(M)]\}$.

For ease of notation, we sometimes define $M(T[n], T'[n])$ also as $M(T[n])$, where we separately describe how the counterexamples $T'(n)$ are presented to the conjecture of M on input $T[n]$.

One can similarly define **NCIt**-learning, where the learner's output depends only on the previous conjecture, the latest positive data, and the counterexample provided.

Definition 8. [JK07] (a) M is iterative (for learning from positive data and negative counterexamples), iff there exists a partial recursive function F such that, for all T, T' and n, $M(T[n+1], T'[n+1]) = F(M(T[n], T'[n]), T(n), T'(n))$. Here $M(\lambda, \lambda)$ is some predefined constant.

(b) M **NCIt**-*identifies* $\mathcal{L}$, iff M is iterative, and M **NCEx**-identifies $\mathcal{L}$.

(c) $\mathbf{NCIt} = \{\mathcal{L} : (\exists M)[M\ \mathbf{NCIt}\text{-identifies } \mathcal{L}]\}$.

We will often identify F above with M (that is use $M(p, x, y) = F(p, x, y)$ to describe $M(T[n+1], T'[n+1])$, where $p = M(T[n], T'[n])$ and $x = T(n), y = T'(n)$). This is for ease of notation.

One should also note that the **NCIt** model is equivalent to allowing finitely many subset queries (with counterexamples for the answer "no") in iterative learning.

One can extend the above definition to **NCIt**-learning with m-feedback or m-memory, by allowing the learner M up to m queries about whether some element x has appeared in the previous text or allowing the learner M to remember up to m elements of the past data. The resulting criteria are called **NCIt**-learning with m-feedback and **NCIt**-learning with m-memory, respectively. The resulting learners are called m-feedback **NCIt**-learner (or **NCIt**-learners using m-feedback) and m-memory bounded **NCIt**-learner (or **NCIt**-learner using m-memory) respectively.

It follows from the definition that **NCIt**-learning is contained in **NCIt**-learning using m-feedback and **NCIt**-learning using m-memory, which, in turn, are contained in **NCEx**.

3 Some General Effects of Additional Information on NCIt-learning

In this section, we look at some known capabilities of **NCIt**-learners (established in [JK07]) and explore whether they hold when a learner has access to additional information.

It was shown in [JK07] that capabilities of **NCIt**-learners exceed capabilities of **InfIt**-learners. In this section, we show that if an **InfIt**-learner can store just one element seen so far, or can use just one feedback query, then it can sometimes learn more than any **NCEx**-learner (which can memorize the whole input seen so far)! However, total **InfIt**-learners having access to the maximal element still can be simulated by **NCIt**-learners having access to the maximal element.

An important result established in [JK07] is that **NCIt**-learners can infer any *indexed* class of recursive languages. However, it is also shown in [JK07] that, surprisingly, such **NCIt**-learners cannot learn indexed classes *class-preservingly* (cf.[ZL95]), that is, using a numbering of languages containing exactly the target class (and no other languages). Still class-preserving learnability is important, as any natural hypotheses space for an indexed class is class-preserving. We now show that **NCIt**-learners can learn indexed classes class-preservingly if they have access to the maximal element or the number of elements seen so far. Still, class-preserving learnability might not be possible if an **NCIt**-learner can use n feedback queries.

3.1 Informants versus negative counterexamples

First we show how storing just one element seen so far or using one feedback query can dramatically make an **InfIt**-learner stronger than any learner storing the whole input seen so far, but getting only positive data and negative counterexamples.

Theorem 9. *There exists a class which can be learnt using* 1*-memory bounded (or* 1*-feedback)* **InfIt** *learner, which cannot be learnt by an* **NCEx***-learner.*

Proof. Let $\mathcal{L}_1 = \{L : (\exists e)[L \subseteq \{\langle e, x\rangle : x \in \mathbb{N}\}$ and $W_e = L$ and $(\forall x)[card(\{\langle e, 2x\rangle, \langle e, 2x+1\rangle\} \cap L) \geq 1]]\}$.

Let $\mathcal{L}_2 = \{L : (\exists z, e)[\{\langle e, x\rangle : x \geq 2z + 2\} \subseteq L \subseteq \{\langle e, x\rangle : x \in \mathbb{N}\}$ and $(\forall x < z)[card(\{\langle e, 2x\rangle, \langle e, 2x+1\rangle\} \cap L) \geq 1]$ and $L = \{\langle e, x\rangle : x \in \mathbb{N}\} - (D_z \cup \{\langle e, 2z\rangle, \langle e, 2z+1\rangle\})\}$.

Let $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$. It is easy to verify that $\mathcal{L}$ can be **InfIt**-learnt if 1-feedback query or 1-memory is allowed. A 1-feedback learner initially stores all data (using padding in its hypothesis) until it finds an e such that $(\langle e, x\rangle, 1)$ is in the input for some x. If the stored past data at that point contains $(\langle e, 2z\rangle, 0)$ and $(\langle e, 2z+1\rangle, 0)$, for some z, then the learner makes the conjecture for $\{\langle e, x\rangle : x \in \mathbb{N}\} - (D_z \cup \{\langle e, 2z\rangle, \langle e, 2z+1\rangle\})$, and never changes its mind thereafter. Otherwise, the learner remembers the e (using padding in the hypothesis) and whenever it sees $(\langle e, 2x+b\rangle, 0)$, for some $x \in \mathbb{N}$ and $b \in \{0, 1\}$, it queries $(\langle e, 2x+1-b\rangle, 0)$; if $(\langle e, 2x+1-b\rangle, 0)$ has appeared in past data then the learner conjectures $\{\langle e, x\rangle : x \in \mathbb{N}\} - (D_z \cup \{\langle e, 2z\rangle, \langle e, 2z+1\rangle\})$, and never changes its mind thereafter; otherwise the learner continues to conjecture W_e.

A 1-memory learner can learn $\mathcal{L}$ similarly: after the initial phase (as in 1-feedback learning) for finding the e, it memorizes input $(\langle e, 2x+b\rangle, 0)$, for the maximal such x, where $b \in \{0, 1\}$. If its memory and the new input together are $(\langle e, 2z\rangle, 0)$ and $\langle e, 2z+1\rangle, 0)$, for some z, then it conjectures $\{\langle e, x\rangle : x \in \mathbb{N}\} - (D_z \cup \{\langle e, 2z\rangle, \langle e, 2z+1\rangle\})$, and never changes its mind thereafter; otherwise the learner continues to conjecture W_e.

Now suppose, by way of contradiction, that M **NCEx**-identifies $\mathcal{L}$. Then, by Kleene's recursion theorem [Rog67], there exists an e such that W_e may be defined as follows.

Initially, W_e contains $\langle e, 0\rangle$. Let W_e^s denote W_e defined by the beginning of stage s. Let σ_0 be a sequence with content $\{\langle e, 0\rangle\}$. Let σ_s denote the initial segment constructed before stage s (it will be the case that $W_e^s = \text{content}(\sigma_s)$). $f_s(i)$ will be a function denoting counterexamples given to the learner M on its conjecture i (in the simulation at stage s). It will be the case that range of f_s (except for #) is a subset of E_s — which we will bar from belonging to W_e to maintain the validity of any negative counterexamples given. Initially, $f_0(i) = \#$, for all i, and $E_0 = \emptyset$. It will be the case that, for all x, at most one of $\langle e, 2x\rangle$ and $\langle e, 2x+1\rangle$ will belong to E_s (this is ensured by adding $\langle e, 2x+1-b\rangle$ to W_e whenever we add $\langle e, 2x+b\rangle$ to E_s, where $b \in \{0, 1\}$). Go to stage 0.

Stage s

1. Simulate M by giving counterexamples according to f_s. Dovetail steps 2 and 3 until one of them succeeds. If step 2 succeeds before step 3, if ever, then go to step 4. If step 3 succeeds before step 2, if ever, then go to step 5. Here we assume that step 2 has some priority in the sense that if it can succeed for $t \leq s$, then it succeeds first, with σ being the shortest for which such $t \leq s$ exists.
2. Search for a $\sigma \subseteq \sigma_s$ and a t, such that $W_{M(\sigma),t} - \text{content}(\sigma_s) \neq \emptyset$ and $min(W_{M(\sigma),t} - \text{content}(\sigma_s)) \neq f_s(M(\sigma))$.
3. Search for a $\tau \supseteq \sigma_s$ such that $\text{content}(\tau) \subseteq \{\langle e, x\rangle : x \in \mathbb{N}\} - E_s$ such that $M(\tau) \neq M(\sigma_s)$.
4. Let

   $f_{s+1}(M(\sigma)) = min(W_{M(\sigma),t} - \text{content}(\tau))$,
   $f_{s+1}(i) = f_s(i)$, for $i \neq M(\sigma)$,
   $E_{s+1} = E_s \cup \{f_s(M(\sigma))\}$, and
   For all $\langle e, 2x+b\rangle \in E_s$, where $x \in \mathbb{N}, b \in \{0, 1\}$, enumerate $\langle e, 2x+1-b\rangle$ in W_e.
   Let W_e^{s+1} be W_e enumerated upto now.
   Let σ_{s+1} be an extension of σ_s such that $\text{content}(\sigma_{s+1}) = W_e^{s+1}$.

Go to stage $s+1$.

5. Let $\sigma_{s+1} = \tau$.
   Enumerate content(σ_{s+1}) in W_e.
   Let $W_e^{s+1} = \text{content}(\sigma_{s+1})$, $E_{s+1} = E_s$, $f_{s+1} = f_s$.
   Go to stage $s+1$

End stage s

Now if there are infinitely many stages, then $W_e \in \mathcal{L}_1$, and $T = \bigcup_s \sigma_s$ is a text for W_e. Suppose $M(T)$ converges. Then for large enough stage s, step 2 would not succeed anymore (as the least counterexamples would have been found by then). Thus, step 3 succeeds infinitely often, and M does not converge on T, a contradiction to the assumption that $M(T)$ converges.

Thus, there are only finitely many stages. Suppose stage s starts but does not end. Hence the counterexamples as in f_s on initial segments of τ (as in stage s) are correct. Moreover, $W_{M(\sigma_s)} \subseteq \text{content}(\sigma_s)$ or $W_{M(\sigma_s)}$ contains an element in E_s. Furthermore, M does not change its mind on any extension τ of σ such that content$(\tau) \subseteq \{\langle e, x\rangle : x \in \mathbb{N}\} - E_s$. Let z be such that $D_z \supseteq E_s$ and $(D_z - E_s) \cap \{\langle e, x\rangle : x \in \mathbb{N}\} = \emptyset$, and $z > max(\{x : \langle e, x\rangle \in E_s\})$. Now, M fails on the language $\{\langle e, x\rangle : x \in \mathbb{N}\} - (D_z \cup \{\langle e, 2z\rangle, \langle e, 2z+1\rangle\})$. □

Still, as the next theorem demonstrates, *total* **InfIt**-learners (that is, the ones that are defined on all, even, possibly, non-valid inputs — that is, data which does not represent a possible previous conjecture, a new input element, and the maximal element possible in a valid learning process for a language in the class being learnt) can be simulated by **NCIt**-learners if both have access to the maximal element. For learning from informants, the maximal element present in the input is the maximal y such that $(y, 0)$ or $(y, 1)$ is present in the input given so far.

Theorem 10. *Any class which is* **InfIt** *learnable using the maximal element by a total learner is also* **NCIt***-learnable using maximal element.*

Proof. Suppose an **InfIt** learner M using the maximal element, for a class $\mathcal{L}$ is given. Below $M(i, (w, b), x)$ denotes output of M when the previous hypothesis is i, (w, b) is the current input, and x is the maximal element such that (x, b), for some $b \in \{0, 1\}$, has been in the input provided to M so far).

On input text T for a language $L \in \mathcal{L}$, the aim of M' is to search for an initial information segment σ such that

(A) For all $w \in L$, $w \geq |\sigma|$, $M(M(\sigma), (w, 1), w) = M(\sigma)$.

(B) $\{w \geq |\sigma| : M(M(\sigma), (w, 0), w) \neq M(\sigma)\} \subseteq L$.

Note that any such σ would imply $M(\sigma)$ is a grammar for the input language L. σ satisfying (A) and (B) above are called good (for L).

The hypothesis of M' is of form $P(\sigma)$ or $R(\sigma)$, or $Q(w, \sigma, S, m)$, where $w, m \in \mathbb{N}$, and S is a finite set. $Q(w, \sigma, S, m)$ is a grammar for $\{w\}$. $P(\sigma)$ is a grammar for $W_{M(\sigma)}$. $R(\sigma)$ is a grammar for $\{w \geq |\sigma| : M(M(\sigma), (w, 0), w) \neq M(\sigma)\}$. In all the above conjectures, σ would be some appropriate initial segment of the canonical informant for the input language. $R(\sigma)$ is a conjecture of the learner when it is testing whether clause (B) above holds. $P(\sigma)$ is the hypothesis which the learner makes in a situation when it thinks that it has a potentially good conjecture for the input language (which satisfies (B) and satisfies (A) for the input seen so far). When the learner does not have a good conjecture, it needs to determine the membership of some w in the input language — $Q(w, \sigma, S, m)$ is used for this purpose. During this membership determining phase, S would be a finite set of elments from the input, which the learner remembers separately, in order not to forget them due to memory limitation.

Initially, $M(\lambda) = R(\lambda)$.

For other inputs M' behaves as follows.

If the previous conjecture is $R(\sigma)$ and the new input is w, and counterexample is not #, then M' outputs $Q(|\sigma|, \sigma, \emptyset, max(\{m, |\sigma|+1\}))$, where m is the maximal element seen so far. Similarly, if $w \neq \#$ and $M(M(\sigma), (w, 1), w) \neq M(\sigma)$, then M' outputs $Q(|\sigma|, \sigma, \emptyset, max(\{m, |\sigma|+1\}))$, where m is the maximal element seen so far. Otherwise, M' outputs $P(\sigma)$.

If the previous conjecture is $P(\sigma)$, and the new input is w and counterexample is not #, then M' outputs $Q(|\sigma|, \sigma, \emptyset, max(\{m, |\sigma|+1\}))$, where m is the maximal element seen so far. Similarly, if $w \neq \#$, and $M(M(\sigma), (w, 1), w) \neq M(\sigma)$, then M' outputs $Q(|\sigma|, \sigma, \emptyset, max(\{m, |\sigma| + 1\}))$, where m is the maximal element seen so far. Otherwise, M' continues with the hypothesis $P(\sigma)$.

If the previous conjecture is $Q(x, \sigma, S, m)$ and the new input is w and $x < m$, then M' outputs $Q(x + 1, \sigma \diamond (x, b), S \cup \{w\} - \{\#\}, m)$, where b is 1 if counterexample is #, and 0 otherwise. If $x = m$, then M' additionally checks if $M(\sigma \diamond (x, b)) \neq M(M(\sigma \diamond (x, b), (y, 1), y)$ for some $y \in S \cup \{w\} - \{\#\}$, $y \geq |\sigma|$; if so then M' outputs $Q(x+1, \sigma \diamond (x, b), \emptyset, max(S \cup \{w\} - \{\#\}) + m + 1)$. Otherwise, M' outputs $R(\sigma \diamond (x, b))$.

Note that all σ used in the hypothesis of the form $R(\sigma)$, $Q(w, \sigma, S, m)$, and $P(\sigma)$ are initial segments of a canonical information segment for the input language. Also note that either the hypotheses of the learner M' converge, or σ (as in the hypotheses) is unbounded in length.

Also note that there exists a good τ which is an initial segment of the canonical sequence, as M learns the input language on the canonical informant. Fix one such good segment τ for the input language. Furthermore, all initial segments of the canonical sequence which extend τ are also good. It is then easy to verify that once the σ, as in the hypotheses of M', extends τ, it will eventually output $P(\sigma')$ for some extension σ' of σ such that σ' is an initial segment of the canonical segment of the input language (for this, note that if the previous hypothesis of M' was $R(\sigma)$ or $P(\sigma)$, then it will next output $P(\sigma)$ and will not change its mind thereafter; if its previous hypothesis is of the form $Q(i, \sigma, S, m)$, then once value of i reaches m, the learner will output $R(\sigma')$ for some extension σ' of σ, where σ' is an initial segment of the canonical sequence for the input language). It follows that eventually the learner will output $P(\sigma)$ where σ is good, and then never changes its mind. It thus follows that M' learns L. $\square$

3.2 Indexed Families

Unlike the case of **NCIt**-learnability (without access to additional information), class-preserving learnability of indexed classes can be achieved if an **NCIt**-learner has access to the maximal element or the number of elements seen.

Theorem 11. *(a) Every indexed family can be* **NCIt***-identified (using a class preserving hypothesis space) given the maximal element seen so far.*

(b) Every indexed family can be **NCIt**-*identified (using a class preserving hypothesis space) given the number of elements seen so far.*

Proof. (a) Suppose $\mathcal{L}$ is an indexed family, and $L_0, L_1, \ldots$ is its listing where $x \in L_i$ can be effectively determined in x and i. Let $L_i[m]$ denote $\{x \in L_i : x \leq m\}$. The conjectures of the learner would be of the form: $p(j, S, X)$, where $p(j, S, X)$ is a grammar for L_j, and S, X are finite sets with some properties.

Suppose T is an input text for a language L, where $T(n) = x_n$. Inductively, if $p(j_n, S_n, X_n)$ is output (after $T[n]$ has been seen), then the following invariants will hold.

(A) for each $j \in S_n$, $L_j \subseteq L$, and $X_n \subseteq L$.

(B) content$(T[n]) \subseteq X_n \cup \bigcup_{j \in S_n} L_j$,

(C) For all $j < j_n$, $L_j \neq L$.

(D) If $j_n \notin S_n$, then either $n = 0$ or $j_n = j_{n-1} + 1$.

(E) $X_n \subseteq X_{n+1}$, $S_n \subseteq S_{n+1}$, $j_n \leq j_{n+1}$.

Initially, $M(\lambda) = (0, \emptyset, \emptyset)$. The learner on the input $p(j_n, S_n, X_n)$ and the new element x_n, the counterexample y_n, and the maximal element m seen so far, does the following:

(i) If $y_n = \#$, then $S_{n+1} = S_n \cup \{j_n\}$; otherwise $S_{n+1} = S_n$.

(ii) If $(X_n \cup \{x_n\} \cup \bigcup_{j \in S_n} L_j[m]) - \{\#\} \subseteq L_{j_n}$, and $y = \#$, then $j_{n+1} = j_n$, $X_{n+1} = X_n$. Otherwise, $j_{n+1} = j_n + 1$ and $X_{n+1} = X_n \cup \{x_n\} - \{\#\}$.

It is easy to verify that the invariants are satisfied. Furthermore, j_n never goes beyond the minimal grammar for L (see invariant (C)). Thus, the sequence of j_n converges, as well as S_n and X_n converge (as $X_{n+1} \neq X_n$ implies $j_{n+1} \neq j_n$, and $S_n \subseteq \{j : j \leq j_n\}$, and using invariants (D) and (E)). Moreover, the last conjecture is correct by (A) and (B), and using $(X_n \cup \{x_n\} \cup \bigcup_{j \in S_n} L_j[m]) - \{\#\} \subseteq L_{j_n}$ from clause (ii) (as there is no further mind change).

(b) Only change is in (ii) above which is replaced by: (m below denotes the number of elements seen so far by the learner)

(ii) If the first m elements in $(X_n \cup \{x_n\} \cup \bigcup_{j \in S_n} L_j) - \{\#\}$ are included in L_{j_n}, and $y = \#$, then $j_{n+1} = j_n$, $X_{n+1} = X_n$. Otherwise, $j_{n+1} = j_n + 1$ and $X_{n+1} = X_n \cup \{x_n\} - \{\#\}$.

The rest of the proof is similar to the part (a), and we omit the details. □

Still, any n feedback queries might not help to achieve class-preserving learnability of indexed classes by **NCIt**-learners.

Theorem 12. *There exists an indexed family which cannot be learnt by an* **NCIt**-*learner with n-feedback using a class preserving hypothesis space.*

Proof. (brief sketch) In [JK07], the corresponding result is shown for **NCIt**-learning (Theorem 32). The proof for the current theorem can be done in essentially the same way, except that the learner might be a feedback learner, so we need to answer its queries appropriately on the σ/τ's etc. that is built in Theorem 32 of [JK07]. When we add "a" and "b" as in there, we can do as there. However, when we add "c", we need to choose the "c" which has not been queried on prefixes of input $\tau\#$ (with the appropriate counterexamples $\tau'\#$) (see step 5 in Theorem 32 proof in [JK07]); there we were using $w + 1$ as c. For the current theorem, we take some larger w, which has not been queried by the learner). This is so that when one considers σc as not making a mind change in step 6, then we can argue that $\sigma c\gamma\#^\infty$ would also not make a mind change (where $\sigma \diamond \gamma = \tau$)

The remaining part of the proof of Theorem 32 in [JK07] can stand as it is. □

4 Hierarchy of n-Feedback and n-Memory Learners

In this section, we show that, in the context of **NCIt**-learnability, $n+1$ stored input elements seen and $n+1$ feedback queries provide more capability than n stored input elements seen and, respectively, n feedback queries. Note that, on the negative sides of both results, neither **NCIt**-learners storing just up to n elements seen, nor **NCIt**-learner using just up to n feedback queries can be helped even if they have access to the maximal element and the number of elements seen so far. On the other hand, learners witnessing the positive sides of both results do not need access to negative counterexamples (refuting conjectures containing data in excess of the target language).

Theorem 13. *Fix $n \in \mathbb{N}$. There exists a class $\mathcal{L}$ such that*

(a) $\mathcal{L}$ can be iteratively learnt by a $n+1$-feedback learner.

(b) $\mathcal{L}$ cannot be **NCIt***-learnt using n-feedback queries even if the maximal element and the number of elements in the input seen so far is given to the learner as additional information.*

Proof. Let $\mathcal{L}_1 = \{L : (\exists e)[\emptyset \subset L \subseteq \{\langle e, j, x\rangle : j, x \in \mathbb{N}\}$ and $W_e = L$ and for all $x, card(\{j : \langle e, j, x\rangle \in L, j \geq 1\}) \leq 1]\}$.

$\mathcal{L}_2 = \{L : (\exists e, x, j, j' : 1 \leq j < j' \leq n+2)[W_e \in \mathcal{L}_1,\ x > max(\{x' : (\exists j' \geq 1)[\langle e, j', x'\rangle \in W_e]\})$, and $L = W_e \cup \{\langle e, j, x\rangle, \langle e, j', x\rangle\}]\}$.

Let $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$.

It is easy to verify that $\mathcal{L}$ can be learnt using $n+1$ feedback queries. The learner can easily determine e such that the input language is a subset of $\{\langle e, j, x\rangle : j, x \in \mathbb{N}\}$. Initially, on the first non # input, the learner outputs e (padded so that the learner can recognize that it is in this phase). For every further input of the form $\langle e, j, x\rangle$ such that $1 \leq j \leq n+2$, the learner queries if the earlier data contains any of the element in $\{\langle e, j', x\rangle : 1 \leq j' \leq n+2, j \neq j'\}$. If so, then the learner outputs a grammar for $W_e \cup \{\langle e, j, x\rangle, \langle e, j', x\rangle\}$, for the $j' \neq j$ such that $\langle e, j', x\rangle$ belongs to the input seen so far, and does not change its mind any further.

Now suppose, by way of contradiction, that a learner M **NCIt** learns $\mathcal{L}$ using n feedback queries (along with the maximal element as well as the number of elements seen in the input so far). Then, by Kleene's recursion theorem [Rog67], there exists an e such that W_e may be defined as follows.

Initially, W_e contains $\langle e, 0, 0\rangle$. Let W_e^s denote W_e defined by the beginning of stage s. Let σ_0 be a sequence with the content $\{\langle e, 0, 0\rangle\}$. Let σ_s denote the initial segment constructed before stage s (it will be the case that $W_e^s = \text{content}(\sigma_s)$). $f_s(i)$ will be a function denoting counterexamples given to the learner M on its conjecture i (in the simulation at stage s). It will be the case that the range of f_s (except for #) is a subset of E_s — which we will bar from belonging to W_e to maintain the validity of any negative counterexamples given. Initially, $f_0(i) = \#$, for all i, and $E_0 = \emptyset$. Let x_s denote the least number such that $W_e^s \cup E_s \subseteq \{\langle e, j, x\rangle : x < x_s\}$. Go to stage 0.

Stage s

1. Let $m > x_s$ be a large enough number such that $\langle e, 0, m\rangle > max(\text{content}(\sigma_s) \cup \{\langle e, j, x\rangle : x_s \leq x \leq x_s + 1 \text{ and } 1 \leq j \leq n + 2\})$.
   Enumerate $\langle e, 0, m\rangle$ in W_e, and let $\tau = \sigma_s \diamond \langle e, 0, m\rangle$.
2. Simulate M by giving counterexamples according to f_s. Dovetail steps 3 and 4 until one of them succeeds. If step 3 succeeds before step 4, if ever, then go to step 5. If step 4 succeeds before step 3, if ever, then go to step 6. Here we assume that step 3 has some priority in the sense that if it can succeed for $t \leq s$, then it succeeds first, with σ being the shortest for which such $t \leq s$ exists.
3. Search for a $\sigma \subseteq \tau$ and a t, such that $W_{M(\sigma),t} - \text{content}(\tau) \neq \emptyset$ and $min(W_{M(\sigma),t} - \text{content}(\tau)) \neq f_s(M(\sigma))$.
4. Search for a j, j', x such that $1 \leq j, j' \leq n + 2$, $j \neq j'$, $x_s \leq x \leq x_s + 1$ and
   (a) $M(\tau \diamond \langle e, j, x\rangle)\downarrow \neq M(\tau)\downarrow$ or
   (b) $M(\tau \diamond \langle e, j, x\rangle\langle e, j', x\rangle)\downarrow \neq M(\tau)\downarrow$, where M on previous conjecture $M(\tau \diamond \langle e, j, x\rangle)$ and new data $\langle e, j', x\rangle$) did not query $\langle e, j, x\rangle$.
5. Let
   $\sigma_{s+1} = \tau$,
   $f_{s+1}(M(\sigma)) = min(W_{M(\sigma),t} - \text{content}(\tau))$,
   $f_{s+1}(i) = f_s(i)$, for $i \neq M(\sigma)$,
   $W_e^{s+1} = W_e$ enumerated until now.
   $E_{s+1} = E_s \cup \{f_s(M(\sigma))\}$, and
   $x_{s+1} =$ the least number such that $W_e^{s+1} \cup E_{s+1} \subseteq \{\langle e, j, x\rangle : x < x_{s+1}\}$.
   Go to stage $s + 1$.
6. In case (a) let $\sigma_{s+1} = \tau \diamond \langle e, j, x\rangle$.
   In case (b) let $\sigma_{s+1} = \tau \diamond \langle e, j'', x'\rangle\langle e, j', x\rangle$, where j'', x' is such that $1 \leq j'' \leq n + 2$ and $x \neq x'$, $x_s \leq x' \leq x_s + 1$ and $\langle e, j'', x'\rangle$ is not queried by $M(\tau \diamond \langle e, j, x\rangle\langle e, j', x\rangle)$.
   Let $W_e^{s+1} = \text{content}(\sigma_{s+1})$ and update $E_{s+1} = E_s$, $f_{s+1} = f_s$ and $x_{s+1} =$ the least number such that $W_e^{s+1} \cup E_{s+1} \subseteq \{\langle e, j, x\rangle : x < x_{s+1}\}$.
   Go to stage $s + 1$

End stage s

Now if there are infinitely many stages, then $W_e \in \mathcal{L}_1$, and $T = \bigcup_s \sigma_s$ is a text for W_e. Suppose $M(T)$ converges. Then, for large enough stage s, step 3 would not succeed anymore (as the least counterexamples would have been found by then). Thus, step 4 succeeds infinitely often, and M does not converge on T, a contradiction to the assumption that $M(T)$ converges.

Thus, there are only finitely many stages. Suppose stage s starts but does not end. Hence the counterexamples as in f_s on initial segments of τ (as in stage s) are correct. Let $\langle e, j', x_s\rangle$, $\langle e, j, x_s\rangle$, $j \neq j'$ with $1 \leq j, j' \leq n + 2$, be such that M on the previous conjecture $M(\tau \diamond \langle e, j, x_s\rangle)$ and new input $\langle e, j', x_s\rangle$, does not query $\langle e, j, x_s\rangle$. Note that $M(\tau)\downarrow = M(\tau \diamond \langle e, j, x_s\rangle\langle e, j', x_s\rangle)\downarrow$, and $W_{M(\tau)}$ either does not enumerate any element outside content(τ), or the least such element is $f_s(M(\tau))$ (by non-success of step 3), where $f_s(M(\tau))$ is different from $\langle e, j, x_s\rangle$, $\langle e, j', x_s\rangle$ (by definition of x_s). Thus, M does not **NCIt** learn with n feedback queries the language $W_e \cup \{\langle e, j, x_s\rangle, \langle e, j', x_s\rangle\}$, which belongs to $\mathcal{L}_2$. $\square$

Theorem 14. *Let $n \in \mathbb{N}$. There exists a $\mathcal{L}$ such that*

(a) $\mathcal{L}$ can be iteratively learnt using $(n+1)$-bounded-memory.

(b) $\mathcal{L}$ cannot be learnt by a **NCIt***-learner using n-memory, even if the learner is given the number of elements and the maximal element seen so far as additional information.*

Proof. Let $\mathcal{L}_1 = \{L : (\exists e)[\emptyset \subset L \subseteq \{\langle e, j, x\rangle : j, x \in \mathbb{N}\}$ and $W_e = L$ and for all x, $[card(\{j : \langle e, j, x\rangle \in L, j \geq 1\}) \leq n+1$ or $[card(\{j : \langle e, j, x\rangle \in L, j \geq 1\}) = n+2$ and $\sum_{\langle e,j,x\rangle \in L} j$ is a prime number]]]$\}$.

Let $\mathcal{L}_2 = \{L : (\exists e, x)[W_e \in \mathcal{L}_1,\ x > max(\{x' : \langle e, j, x'\rangle \in W_e, j' \geq 1\})$ and $[card(\{j : \langle e, j, x\rangle \in L, j \geq 1\}) = n+2$ and $\sum_{\langle e,j,x\rangle \in L} j$ is not a prime number] and $L = W_e \cup \{\langle e, j, x\rangle : j \geq 1, \langle e, j, x\rangle \in L\}]\}$.

Let $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$.

It is easy to verify that $\mathcal{L}$ can be learnt using $n+1$ memory. The learner can easily determine e such that the input language is a subset of $\{\langle e, j, x\rangle : j, x \in \mathbb{N}\}$. Initially, on the first non # input, the learner outputs e (padded so that it can recognize that it is in this phase). The memory of the learner always consists of all elements of the form $\langle e, j, x\rangle$ seen so far (unless it exceeds $n+1$) where x is the maximal such that for some j', $\langle e, j', x\rangle$ is seen in the input so far. If and when a learner sees an input $\langle e, j, x\rangle$, $j \geq 1$, such that it has memorized $n+1$ distinct elements from the set $\{\langle e, j', x\rangle : j' \geq 1, j' \neq j\}$, and $j + \sum_{\langle e,j',x\rangle \in \text{memory}} j'$ is not a prime number, then it outputs $W_e \cup \{\langle e, j, x\rangle\} \cup \{\langle e, j', x\rangle : \langle e, j', x\rangle$ in memory $\}$, and never changes its mind thereafter.

Now suppose, by way of contradiction, that a learner M **NCIt** learns $\mathcal{L}$ using n-bounded-memory (along with the maximal element as well as the number of elements seen in the input so far). Then, by Kleene's recursion theorem [Rog67], there exists an e such that W_e may be defined as follows.

Initially, W_e contains $\langle e, 0, 0\rangle$. Let W_e^s denote W_e defined by the beginning of stage s. Let σ_0 be a sequence with content $\{\langle e, 0, 0\rangle\}$. Let σ_s denote the initial segment constructed before stage s (it will be the case that $W_e^s = \text{content}(\sigma_s)$). $f_s(i)$ will be a function denoting counterexamples given to the learner M on its conjecture i (in the simulation at stage s). It will be the case that the range of f_s (except for #) is a subset of E_s — which we will bar from belonging to W_e to maintain the validity of any negative counterexamples given. Initially, $f_0(i) = \#$, for all i, and $E_0 = \emptyset$. Let x_s denote the least number such that $W_e^s \cup E_s \subseteq \{\langle e, j, x\rangle : x < x_s\}$. Go to stage 0.

Stage s

1. Let $w > n+1$ be a large enough number such that $(w + card(\text{content}(\sigma_s)) + 2)^n < \binom{w}{n+1}$.
   Let w' be such that for all distinct $c, c' \leq (n+1) * 2^w$, there exists a p with $2^w < p \leq w'$ such that $c+p$ is a prime, but $c'+p$ is not a prime. Let $m > x_s$ be such that $\langle e, 0, m\rangle > max(\text{content}(\sigma_s) \cup \{\langle e, j, x_s\rangle : 1 \leq j \leq w'\})$.
   Enumerate $\langle e, 0, m\rangle$ in W_e, and let $\tau = \sigma_s \diamond \langle e, 0, m\rangle$.
2. Simulate M by giving counterexamples according to f_s. Dovetail steps 3 and 4 until one of them succeeds. If step 3 succeeds before step 4, if ever, then go to step 5. If step 4 succeeds before step 3, if ever, then go to step 6. Here we assume that step 3 has some priority in

the sense that if it can succeed for $t \leq s$, then it succeeds first, with σ being the shortest for which such $t \leq s$ exists.

3. Search for a $\sigma \subseteq \tau$ and a t, such that $W_{M(\sigma),t} - \text{content}(\tau) \neq \emptyset$ and $min(W_{M(\sigma),t} - \text{content}(\tau)) \neq f_s(M(\sigma))$.
4. Search for a τ' such that either
   (a) $\text{content}(\tau') - \text{content}(\tau) \subseteq \{\langle e, j, x_s\rangle : 1 \leq j \leq w'\}$, and $card(\text{content}(\tau') - \text{content}(\tau)) \leq n+1$, and $M(\tau') \neq M(\tau)$ or
   (b) $\text{content}(\tau') - \text{content}(\tau) \subseteq \{\langle e, j, x_s\rangle : 1 \leq j \leq w'\}$, and $card(\text{content}(\tau') - \text{content}(\tau)) = n+2$, and $\sum_{\langle e,j,x\rangle \in \text{content}(\tau') - \text{content}(\tau)} j$ is a prime number, and $M(\tau') \neq M(\tau)$.
5. Let

   $\sigma_{s+1} = \tau$,
   $f_{s+1}(M(\sigma)) = min(W_{M(\sigma),t} - \text{content}(\tau))$,
   $f_{s+1}(i) = f_s(i)$, for $i \neq M(\sigma)$,
   $W_e^{s+1} = W_e$ enumerated until now.
   $E_{s+1} = E_s \cup \{f_s(M(\sigma))\}$, and
   $x_{s+1} =$ the least number such that $W_e^{s+1} \cup E_{s+1} \subseteq \{\langle e, j, x\rangle : x < x_{s+1}\}$.

   Go to stage $s+1$.
6. Let $\sigma_{s+1} = \tau'$,
   Let $W_e^{s+1} = \text{content}(\sigma_{s+1})$ and update $E_{s+1} = E_s$, $f_{s+1} = f_s$ and $x_{s+1} =$ the least number such that $W_e^{s+1} \cup E_{s+1} \subseteq \{\langle e, j, x\rangle : x < x_{s+1}\}$.
   Go to stage $s+1$

End stage s

Now if there are infinitely many stages, then $W_e \in \mathcal{L}_1$, and $T = \bigcup_s \sigma_s$ is a text for W_e. Suppose $M(T)$ converges. Then for large enough stage s, step 3 would not succeed anymore (as the least counterexamples would have been found by then). Thus, step 4 succeeds infinitely often, and M does not converge on T, a contradiction to the assumption that $M(T)$ converges.

Thus, there are only finitely many stages. Suppose stage s starts but does not end. Hence the counterexamples as in f_s on initial segments of τ (as in stage s) are correct. Let the parameters below be as in stage s. For each set S of $n+1$ elements in $\{2^j : 1 \leq j \leq w\}$, let τ_S' be such that $\tau \subseteq \tau_S$ and $\text{content}(\tau_S) - \text{content}(\tau) = S$. Let $q_S = \sum_{\langle e,j,x\rangle \in S} j$. Let S, S' be distinct sets of $n+1$ elements in $\{2^j : 1 \leq j \leq w\}$ such that memory of $M(\tau_S)$ and memory of $M(\tau_{S'})$ are same. Note that there exist such distinct S, S' by hypothesis on w. Furthermore, $q_S \neq q_{S'}$. Now, let p be such that $2^w < p \leq w'$, and $q_S + p$ is a prime number, but $q_{S'} + p$ is not a prime number. Then, as M did not change its mind on $\tau_S \diamond \langle e, p, x_s\rangle^\infty$, M does not change its mind on $\tau_{S'} \diamond \langle e, p, x_s\rangle$ either. Thus, $M(\tau_{S'} \diamond \langle e, p, x_s\rangle^\infty) = M(\tau)$, and $W_{M(\tau)}$ either does not enumerate any element outside content(τ), or the least such element is $f_s(M(\tau))$ (by non-success of step 3), which is different from any of $\langle e, p, x_s\rangle$ (by definition of x_s). Thus, M does not converge to a grammar for $\text{content}(\tau_{S'}) \cup \{\langle e, p, x_s\rangle\}$, which is a member of $\mathcal{L}_2$. □

5 Advantages of Different Types of Additional Information Over Other Types

In these section we study tradeoffs between different types of additional information in the context of **NCIt**-learnability.

5.1 Comparison of Feedback and Memory Bounded Learning

Results of this subsection significantly strengthen corresponding results given in [CJLZ99]. Namely, they demonstrate that, in the context of **NCIt**-learnability, just one stored inputted element can provide more than any n feedback queries (even if, in addition, the learner has access to the maximal element and the number of elements seen so far), and, conversely, one feedback query can do more than any n stored input elements seen so far (and, additionally, the maximal element and the number of elements seen so far). Moreover, the iterative learners witnessing the positive sides of these results do not not use negative counterexamples to conjectures containing extra elements.

Theorem 15. *There exists a $\mathcal{S}$ which can be iteratively learnt by a 1-memory bounded learner, but which cannot be* **NCIt***-learnt using n-feedback (even if the learner is given the maximal element and the number of elements in the input so far as additional information).*

Proof. For each n, let $\mathcal{L}_n$ denote the class $\mathcal{L}$ used in the proof of Theorem 13. Let $\mathcal{S} = \bigcup_n \mathcal{L}_n$. Then, by Theorem 13, $\mathcal{S}$ cannot be **NCIt**-learnt using n-feedback. However, $\mathcal{S}$ can be iteratively learnt using 1-memory bound: Initially the learner outputs e (padded), so that $\langle e, \cdot, \cdot \rangle$ is the first element in the input. The learner always memorizes $\langle e, j', x \rangle$, for the largest x such that $\langle e, j', x \rangle$ belongs to the input for some j' (as long as there is only one such corresponding j'). Now if some $\langle e, j, x \rangle$ appears in the input, with $\langle e, j', x \rangle$ in the memory, where $j' \neq j$, then, the learner outputs the grammar for $W_e \cup \{\langle e, j, x \rangle, \langle e, j', x \rangle\}$, and never changes its mind thereafter. □

Theorem 16. *There exists a $\mathcal{L}$ which can be iteratively learnt using 1-feedback learner, but which cannot be* **NCIt***-learnt using n-memory bounded learner (even if the learner is given maximal element and number of elements in the input so far as additional information).*

Proof. Let $\mathcal{L}_1 = \{L : (\exists e)[\emptyset \subset L \subseteq \{\langle e, k, j, x \rangle : k, j, x \in \mathbb{N}\}$ and $W_e = L$ and [for all $k > 0, j > 0, x$, $\langle e, 1, j, x \rangle \in L$ iff $\langle e, k, j, x \rangle \in L]$, and for all x, $[card(\{j : j \geq 1, \langle e, 1, j, x \rangle \in L\}) = 0$ or $[\langle e, 1, j, x \rangle \in L$ for a unique $j \geq 1$, and $\pi_1^2(j) \notin L]]]\}$.

Let $\mathcal{L}_2 = \{L : (\exists e, x)[W_e \in \mathcal{L}_1,\ x > max(\{x' : \langle e, k, j, x' \rangle \in W_e, k \in \mathbb{N}, j \geq 1\})$ and $[\langle e, 1, j, x \rangle \in L$ for a unique $j \geq 1$, and $\pi_1^2(j) \in L$ and $L = W_e \cup D_{\pi_2^2(j)} \cup \{\langle e, k, j, x \rangle : k \geq 1\}]]\}$.

Let $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$.

Using a proof similar to the proof of Theorem 14, it can be shown that $\mathcal{L}$ cannot be **NCIt**-learnt using n-memory (even if the learner is given the maximal element and the number of elements in the input as additional information). However, the above $\mathcal{L}$ is learnable using 1-feedback query, as initially the learner can output conjecture e (actually, a padded version —

to recognize that it is in this phase), and whenever it receives input of the form $\langle e, k, j, x\rangle$, for $j \geq 1$, $k \geq 1$, it can query whether $\pi_1^2(j)$ is in the text seen so far. If so, then it can change its mind to output grammar for $W_e \cup D_{\pi_2^2(j)} \cup \{\langle e, k', j, x\rangle : k' \geq 1\}$, and never changes its mind thereafter. $\square$

Another class (which is simpler to state) which can be used for a proof of Theorem 16 would be:

$\mathcal{L}_1 = \{L : (\exists e)[W_e = L \subseteq \{\langle e, x\rangle : x \in \mathbb{N}\}]$ and for all $x, card(L \cap \{\langle e, 2x\rangle, \langle e, 2x+1\rangle\}) \leq 1\}$.

$\mathcal{L}_2 = \{L : (\exists e, x)[W_e \in \mathcal{L}_1, L = W_e \cup \{\langle e, 2x\rangle, \langle e, 2x+1\rangle\}]\}$.

$\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$.

The diagonalization proof, though similar to the proof of Theorem 14, needs a bit more modification compared to what we needed for the class used in the current proof of Theorem 16.

5.2 Advantages of Using Maximal Element/Number of Elements

Results of this subsection demonstrate various advantages that **NCIt**-learners can get while using the maximal element or/and the number of elements as additional information.

We begin with two simple useful propositions. The following proposition works if memory, instead of being a set, is allowed to be a multiset (thus, a learner can keep the maximal element twice or more, if needed). It is open at present whether this proposition holds if memory is just a set, as in the current paper.

Proposition 17. *Any n-bounded memory learner with the maximal element in the input as additional information can be simulated by an $n+1$-bounded memory learner by using the extra memory for the maximal element seen, as long as the memory of the learner is considered as a multi-set, rather than just a set.*

Proposition 18. *An* **NCIt**-*learner can learn finite sets when given the number of elements or the maximal element and only negative counterexamples (no positive data is needed). Thus, the learner can, for example, do this even when it forgets some of the elements it has received due to some earlier phase.*

Proof. For the number of elements, the learner just cycles through the finite sets of cardinality m (where m is the number of elements in the input), until it gets no counterexample.

For the maximal element, the learner could just check for each x smaller than the maximal element whether x belongs to the input language (this can be done by using conjecture $\{x\}$ — x is in the input language iff the conjecture for $\{x\}$ does not get a counterexample). That will determine the input language. $\square$

Our next result shows that adding access to the maximal element increases learning capability of **NCIt**-learners storing up to n input elements seen so far. Moreover, a learner witnessing the positive side of the result does not need access to negative counterexamples refuting conjectures containing data in excess of the language to be learned.

Theorem 19. *There exists a class $\mathcal{L}$ which can be iteratively learnt by an n-bounded memory learner with maximal element as additional input that cannot be* **NCIt**-*learnt by a n-bounded memory learner.*

Proof. Let $\mathcal{L}_1 = \{L : (\exists e)[W_e = L \text{ and } \emptyset \subset L \subseteq \{\langle e, d, r, z\rangle : r, z \in \mathbb{N}, d > 0\}$ and for all $x, [card(\{j : \langle e, 1, j, x\rangle \in L\}) \leq n]]\}$.

Let $\mathcal{L}_2 = \{L : (\exists e, x', y)[L \subseteq \{\langle e, d, r, z\rangle : d, r, z \in \mathbb{N}\}$, L is finite and $[\langle e, 0, 0, y\rangle\} \in L$ or [for all $x < x'$, $[card(\{j : \langle e, 1, j, x\rangle \in L\}) \leq n$, and $max(\{x : \langle e, 1, j, x\rangle \in L\}) = x'$ and $card(\{j : \langle e, 1, j, x'\rangle \in L\}) = n + 1]]]]\}$.

Let $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$.

It is easy to verify that $\mathcal{L}$ can be learnt using n-memory plus the maximal element. The learner first finds the e such that $\langle e, d, y, z\rangle$ is in the input language for some e. The learner remembers the $\langle e, 1, j, x\rangle$ in its memory, where x is maximized. The learner continues to output e, until it discovers that $\langle e, 0, y, z\rangle$ is in the input for some y, z, or it finds that the input language contains $n+1$ elements of the form $\langle e, 1, j', x\rangle$, for the maximal x such that some element of the form $\langle e, 1, j, x\rangle$ is in the input (for that, it needs to memorize only n of them). It then switches to learning finite sets using the maximal element (see Proposition 18).

Now suppose, by way of contradiction, that a learner M **NCIt** learns $\mathcal{L}$ using at most n bounded memory. Then, by Kleene's recursion theorem [Rog67], there exists an e such that W_e may be defined as follows.

Initially, W_e contains $\langle e, 2, 0, 0\rangle$. Let W_e^s denote W_e defined by the beginning of stage s. Let σ_0 be a sequence with content $\{\langle e, 2, 0, 0\rangle\}$. Let σ_s denote the initial segment constructed before stage s (it will be the case that $W_e^s = \text{content}(\sigma_s)$). $f_s(i)$ will be a function denoting counterexamples given to the learner M on its conjecture i (in the simulation at stage s). It will be the case that the range of f_s (except for #) is a subset of E_s — which we will bar from belonging to W_e to maintain the validity of any negative counterexamples given.

Initially, $f_0(i) = \#$, for all i and $E_0 = \emptyset$. Let x_s denote the least number such that $W_e^s \cup E_s \subseteq \{\langle e, d, j, x\rangle : d, j \in \mathbb{N}, x < x_s\}$. Go to stage 0.

Stage s

1. Simulate M by giving counterexamples according to f_s. Dovetail steps 2 and 3 until one of them succeeds. If step 2 succeeds before step 3, if ever, then go to step 4. If step 3 succeeds before step 2, if ever, then go to step 5. Here we assume that step 2 has some priority in the sense that if it can succeed for $t \leq s$, then it succeeds first, with σ being the shortest for which such $t \leq s$ exists.
2. Search for a $\sigma \subseteq \sigma_s$ and a t, such that $W_{M(\sigma),t} - \text{content}(\sigma_s) \neq \emptyset$ and $min(W_{M(\sigma),t} - \text{content}(\sigma_s)) \neq f_s(M(\sigma))$.
3. Search for a $\tau \supseteq \sigma_s$ such that (a) $\text{content}(\tau) - \text{content}(\sigma_s) \subseteq \{\langle e, d, y, z\rangle : y, z \in \mathbb{N}, d > 0\}$ and for all $x, [card(\{j : \langle e, 1, j, x\rangle \in \text{content}(\tau)\}) \leq n]\}$, and (b) $M(\sigma_s) \neq M(\tau)$.
4. Let

   $\sigma_{s+1} = \sigma_s$,
   $f_{s+1}(M(\sigma)) = min(W_{M(\sigma),t} - \text{content}(\tau))$,
   $f_{s+1}(i) = f_s(i)$, for $i \neq M(\sigma)$,
   $W_e^{s+1} = W_e$ enumerated until now.
   $E_{s+1} = E_s \cup \{f_s(M(\sigma))\}$, and
   $x_{s+1} =$ the least number such that $W_e^{s+1} \cup E_{s+1} \subseteq \{\langle e, d, j, x\rangle : d, j \in \mathbb{N}, x < x_{s+1}\}$.

Go to stage $s + 1$.

5. Let $\sigma_{s+1} = \tau$,

   Let $W_e^{s+1} = \text{content}(\sigma_{s+1})$, $E_{s+1} = E_s$, $f_{s+1} = f_s$, $x_{s+1} =$ the least number such that $W_e^{s+1} \cup E_{s+1} \subseteq \{\langle e, d, j, x\rangle : d, j \in \mathbb{N}, x < x_{s+1}\}$.

   Go to stage $s + 1$

End stage s

Now if there are infinitely many stages, then $W_e \in \mathcal{L}_1$, and $T = \bigcup_s \sigma_s$ is a text for W_e. Suppose $M(T)$ converges. Then for large enough stage s, step 2 would not succeed anymore (as the least counterexamples would have been found by then). Thus, step 3 succeeds infinitely often, and M does not converge on T, a contradiction to the assumption that $M(T)$ converges.

Thus, there are only finitely many stages. Suppose stage s starts but does not end. Hence the counterexamples as in f_s on initial segments of τ (as in stage s) are correct. Let the parameters below be as in stage s. For each set S of n elements, let τ_S be such that $\sigma_s \subseteq \tau_S$ and $\text{content}(\tau_S) - \text{content}(\sigma_s) = \{\langle e, 1, j, x_s\rangle : j \in S\}$.

Now, suppose there exists an S (of size n) such that for infinitely many y, $M(\tau_S \diamond \langle e, 2, 0, y\rangle)$ has the same memory as $M(\tau_S)$. Let $\langle e, 0, 0, z\rangle \notin E_s$. Then, consider M's behaviour on $\tau_S \diamond \langle e, 0, 0, z\rangle^\infty$. If it does not converge, then it does not learn it (where the counterexamples provided are the least ones). Otherwise, let X be the set of counterexamples provided to M on $\tau_S \diamond \langle e, 0, 0, z\rangle^\infty$, and let y be such that $\langle e, 2, 0, y\rangle$ is not in the set of counterexamples provided nor $\langle e, 2, 0, y\rangle \in E_s$, and $M(\tau_S \diamond \langle e, 2, 0, y\rangle)$ has same memory as $M(\tau_S)$. Then, M fails to learn at least one of $\tau_S \diamond \langle e, 0, 0, z\rangle^\infty$ and $\tau_S \diamond \langle e, 2, 0, y\rangle\langle e, 0, 0, z\rangle^\infty$, as it converges to the same conjecture on both.

Otherwise, for all S (of size n), for all but finitely many y, $M(\tau_S \diamond \langle e, 2, 0, y\rangle)$ has different memory than $M(\tau_S)$.

Thus, by taking such S as a subset of size n of $\{1, 2, \ldots, w\}$, for large enough w, we will have that, for two such S, S', for all but finitely many y, memory is same after seeing $\tau_S \diamond \langle e, 2, 0, y\rangle$ or after seeing $\tau_{S'} \diamond \langle e, 2, 0, y\rangle$ (as the memory could either be remembering $\langle e, 2, 0, y\rangle$ or not, and some set of size at most $n - 1$, due to change in memory). Let $j \in S - S'$ and y be large enough (satisfying above) such that $\langle e, 2, 0, y\rangle \notin E_s$. Then, $M(\tau_{S'} \diamond \langle e, 2, 0, y\rangle\langle e, 1, j, x_s\rangle^\infty) = M(\tau_S \diamond \langle e, 2, 0, y\rangle\langle e, 1, j, x_s\rangle^\infty) = M(\sigma_s)$, and $W_{M(\sigma_s)}$ either enumerates an element in E_s, or does not enumerate any element outside $\text{content}(\sigma_s)$. Thus, it fails to identify $\tau_{S'} \diamond \langle e, 2, 0, y\rangle\langle e, 1, j, x_s\rangle^\infty$. □

Our next two results demonstrate that an **NCIt**-learner having access to just the maximal element or the number of elements seen so far can sometimes do more than any **NCIt**-learner using up to n feedback queries. Note, however, that, as the next theorem demonstrates, whereas the maximal element can give more to **NCIt**-learners (in fact, even just iterative learner — not using negative counterexamples to conjectures!) than n feedback queries *and* even the number of elements as additional information, we were not able to achieve a result of similar strength — faring the number of elements seen so far against n feedback queries *and* the maximal element as additional information. Whether it is possible, remains open. A partial solution to this problem (for iterative learners — not using negative counterexamples to conjectures) is given in Theorem 25.

Theorem 20. *There exists a class $\mathcal{L}$ which can be iteratively learnt when the learner is provided the maximal element in the input so far, but the class $\mathcal{L}$ cannot be* **NCIt***-learnt using n-feedback, for any n, even if the learner is given the number of elements in the input as additional information.*

Proof. Let $\mathcal{L}_1 = \{L : (\exists e)[\emptyset \subset L \subseteq \{\langle e, j, x\rangle : j, x \in \mathbb{N}\}$ and $W_e = L$ and for all $x, card(\{j : \langle e, j, x\rangle \in L, j \geq 1\}) \leq 1]\}$.

$\mathcal{L}_2 = \{L : (\exists e, x, j, j' : 1 \leq j < j')[W_e \in \mathcal{L}_1,\ x > max(W_e),\ L = W_e \cup \{\langle e, j, x\rangle, \langle e, j', x\rangle\}]\}$.

Let $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$. Then clearly, $\mathcal{L}$ can be iteratively learnt using the maximal element (as if $\langle e, j, x\rangle$ is the maximal element and one sees $\langle e, j', x\rangle$ in the input, with $j \neq j'$, then we will know that the input language is from $\mathcal{L}_2$). Showing that the above $\mathcal{L}$ cannot be learnt using n-feedback is essentially same as that of the proof of Theorem 13, except that we do not use the element $\langle e, 0, m\rangle$ in the construction. □

Theorem 21. *There exists a $\mathcal{L}$ which can be* **NCIt** *learnt using the number of elements in the input as additional information, but, for all n, $\mathcal{L}$ cannot be* **NCIt***-learnt using n-feedback.*

Proof. Let $\mathcal{L}_1 = \{\{\langle e+1, x\rangle : x \in W_e\} : W_e \neq \emptyset\}$,
$\mathcal{L}_2 = \{D : D$ is finite and $(\exists y)[\langle 0, y\rangle \in D]\}$, and
$\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$.

It is easy to verify that $\mathcal{L}$ can be iteratively learnt using the number of elements in the input. If and when the learner sees $\langle 0, y\rangle$ in the input, for some y, the learner would switch to learning finite sets (see Proposition 18). Otherwise, if the learner sees some input (and only inputs) of the form $\langle e+1, x\rangle$, the learner outputs a grammar for $\{\langle e+1, x\rangle : x \in W_e\}$.

To see that the above class $\mathcal{L}$ cannot be learnt using n-feedback by **NCIt**-learner, suppose, by way of contradiction, that M does so. Then, by Kleene's recursion theorem [Rog67], there exists an e such that W_e may be described as follows.

Initially, W_e contains 0. Let W_e^s denote W_e defined by the beginning of stage s. Let σ_0 be a sequence with content $\{\langle e+1, 0\rangle\}$. Let σ_s denote the initial segment constructed before stage s (it will be the case that content$(\sigma_s) = \{\langle e+1, x\rangle : x \in W_e^s\}$). $f_s(i)$ will be a function denoting counterexamples given to the learner M on its conjecture i (in the simulation at stage s). It will be the case that the range of f_s (except for #) is a subset of E_s — which we will bar from belonging to $\{\langle e+1, x\rangle : x \in W_e\}$ to maintain the validity of any negative counterexamples given. Initially, $f_0(i) = \#$, for all i and $E_0 = \emptyset$. Go to stage 0.

Stage s

1. Simulate M by giving counterexamples according to f_s. Dovetail steps 2 and 3 until one of them succeeds. If step 2 succeeds before step 3, if ever, then go to step 4. If step 3 succeeds before step 2, if ever, then go to step 5. Here we assume that step 2 has some priority in the sense that if it can succeed for $t \leq s$, then it succeeds first, with σ being the shortest for which such $t \leq s$ exists.

2. Search for a $\sigma \subseteq \sigma_s$ and a t, such that $W_{M(\sigma),t} - \text{content}(\sigma_s) \neq \emptyset$ and $min(W_{M(\sigma),t} - \text{content}(\sigma_s)) \neq f_s(M(\sigma))$.
3. Search for a $\tau \supseteq \sigma_s$ such that $\text{content}(\tau) \subseteq \{\langle e+1, x\rangle : x \in \mathbb{N}\} - E_s$, and $M(\tau) \neq M(\sigma_s)$, where the negative counterexamples are given according to f_s.
4. Let

   $\sigma_{s+1} = \sigma_s$,
   $f_{s+1}(M(\sigma)) = min(W_{M(\sigma),t} - \text{content}(\sigma_s))$,
   $f_{s+1}(i) = f_s(i)$, for $i \neq M(\sigma)$,
   $W_e^{s+1} = W_e^s$.
   $E_{s+1} = E_s \cup \{f_s(M(\sigma))\}$, and

   Go to stage $s+1$.
5. In case (a) let $\sigma_{s+1} = \tau$,
   Let $W_e^{s+1} = \text{content}(\sigma_{s+1})$, $E_{s+1} = E_s$, $f_{s+1} = f_s$ and
   Go to stage $s+1$

End stage s

Now if there are infinitely many stages, then $W_e \in \mathcal{L}$, and $T = \bigcup_s \sigma_s$ is a text for W_e. Suppose $M(T)$ converges. Then for large enough stage s, step 2 would not succeed anymore (as the least counterexamples would have been found by then). Thus, step 3 succeeds infinitely often, and M does not converge on T, a contradiction to the assumption that $M(T)$ converges.

Thus, there are only finitely many stages. Suppose stage s starts but does not end. Hence the counterexamples as in f_s on initial segments of σ_s are correct. Let y be such that $\langle 0, y\rangle \notin E_s$. Now consider the behaviour of M on $\sigma_s\langle 0, y\rangle^\infty$, where the counterexamples are the least counterexamples. If the learner does not converge, then it clearly does not identify $\sigma_s\langle 0, y\rangle^\infty$. Otherwise, let X be the set of counterexamples given, and let Y be the set of feedback queries asked on initial segments of the text $\sigma_s\langle 0, y\rangle^\infty$. Note that $X \cup Y$ is finite. Let x be such that $\langle e+1, x\rangle$ does not belong to $E_s \cup X \cup Y \cup \text{content}(\sigma_s)$. Then, M does not identify at least one of $\sigma_s \diamond \langle 0, y\rangle^\infty$, and $\sigma_s \diamond \langle e+1, x\rangle \diamond \langle 0, y\rangle^\infty$, as M's conjectures converge to the same conjecture on both. $\square$

Note that, obviously, the maximal element can always be memorized by a learner and, thus, cannot add more to the learning power of iterative learners than even one memory cell for storing input elements. Therefore, we explore if the number of elements seen so far can give an **NCIt**-learner more advantages than n memorized input elements seen so far. We were able to achieve only a partial solution — showing that the number of elements and the maximal element (or one memory cell) together can provide more power to **NCIt**-learners than n memorized input elements.

Theorem 22. *There exists a class $\mathcal{L}$ such that $\mathcal{L}$ can be* **NCIt***-learnt using 1-memory (or the maximal element) and the number of elements, but cannot be learnt using n feedback or n-memory bounded learner in* **NCIt** *manner, even if it is given the maximal element.*

Proof. We say that e is nice, if $\emptyset \subset W_e \subseteq \{\langle e, j, x\rangle : j, x \in \mathbb{N}\}$, and for all x such that $\langle e, j, x\rangle \in W_e$, $W_e \cap \{\langle e, j, x'\rangle : j \in \mathbb{N}, x' < x\} \subseteq W_{e,x'}$.

Now let $\mathcal{L}_1 = \{L : L = W_e, e$ is nice, and for all x, $card(W_e \cap \{j : \langle e, j, x\rangle \in L\} < x)\}$.

$\mathcal{L}_2 = \{L : (\exists e, x : x > 0)[W_e \in \mathcal{L}_1,\ W_e = W_{e,x}$ and $x > max(W_e)$ and $card(\{j : \langle e, j, x\rangle \in L\}) = x$ and $L = W_{e,x} \cup \{\langle e, j, x\rangle : \langle e, j, x\rangle \in L]\}\}$.

Let $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$.

It is easy to see that $\mathcal{L}$ can be iteratively learnt using the number of elements and 1-memory. The learner initially conjectures e (with padding) such that the first input element is $\langle e, j, x\rangle$ for some j, x. It remembers the largest x (using input element $\langle e, j, x\rangle$) such that some element of the form $\langle e, j, x\rangle$ is in the input. Then, whenever the learner sees input $\langle e, j', x'\rangle$ and the number of elements k, it checks whether $card(W_{e,x}) + x = k$. If so, then it proceeds to identify the input using the technique of Proposition 18.

$\mathcal{L}$ cannot be **NCIt**-learnt using n-feedback or n-memory, as long as only the maximal element in the input is known. This can be shown essentially using the same technique as in the Theorems 13 and 14.

For n-feedback learning, in the diagonalization part, we do similar to Theorem 13, except that we do not need the part which was dealing with the "number of elements"; In stage s, we choose a large enough $x > n + 3$, such that $W_{e,x}$ contains content(σ_s). Then, we place $\langle e, n + 3 + j, x\rangle$ for $j < x - 2$ in W_e (instead of $\langle e, 0, m\rangle$ considered in the proof of Theorem 13). We let τ be an extension of σ_s such that content(τ) = content(σ_s) $\cup \{\langle e, n + 3 + j, x\rangle : j < x - 2\}$. In step 4 we check if there exists a j such that M makes a mind change on $\tau \diamond \langle e, j, x\rangle$, where $1 \le j \le n + 2$. If so, then we proceed as in the construction in Theorem 13. Otherwise, we choose a j, j' such that $1 \le j, j' \le n + 2$, $j \ne j'$ and M on previous conjecture $M(\tau)$ and new element $\langle e, j, x\rangle$ does not query $\langle e, j', x\rangle$. Then, M fails to learn the input $\tau \diamond \langle e, j', x\rangle\langle e, j, x\rangle^\infty$.

Similar modification can be done for memory-bounded learning. □

Can the maximal element give more power to **NCIt**-learners than the number of elements seen so far? The answer to this question is positive — even if the learners using the maximal element are just iterative (not using negative counterexamples to conjectures): it immediately follows from Theorem 20. However, we do not know if the number of elements can give more in the context of **NCIt**-learnability than the maximal element.

Open Problem 23. *Is there a class $\mathcal{L}$ of languages which can be* **NCIt***-learnt using the number of elements as additional information (no memory or feedback) but cannot be* **NCIt***-learnt using n-memory (or n-feedback with the maximal element as additional information or even just the maximal element as additional information)?*

Three following results give partial solutions to the open problems stated above. The first result shows that, under certain natural (and quite weak) assumptions, in the context of **NCIt**-learnability, access to the number of elements can be replaced by access to the maximal element.

Theorem 24. *Suppose $\mathcal{L}$ can be* **NCIt***-identified using the number of elements, where the learner converges on all inputs (here the text input would be from the target class, but the number of elements may sometimes not be valid — we still expect the learner to converge). Then, $\mathcal{L}$ can be* **NCIt***-identified using access to the maximal element.*

Proof. Suppose M, a learner using number of elements is given. We construct M' as follows. Outputs of M' will be of form $P(\sigma, m, S, t)$ (during normal phases) or $Q(m, S, t, i)$ (during discovery phases). $W_{P(\sigma,m,S,t)} = W_{M(\sigma)}$, and $W_{Q(m,S,t,i)} = \{i\}$.

Suppose input language is L.

Intuitively, in a conjecture $P(\sigma, m, S, t)$, $S \subseteq L$ and S will contain all elements $\leq m$ which are contained in the input language L. Furthermore, σ seems like the least stabilizing sequence for M on input S as verified in time t. t is just a counter which increases every time we change phase from normal to discovery. Initially, the conjecture is $P(\lambda, -1, \emptyset, 0)$ (for ease of notation, we use $m = -1$; this is the only negative number used).

Normal phase:

Suppose the previous conjecture of M' was $P(\sigma, m, S, t)$. Suppose the new input element to M' is x and the maximal element seen so far by M' is m'. Now M' simulates M by giving it the previous conjecture as $M(\sigma)$, new element as x, and the number of elements as any number between c and $m' - m + c$, where $c = S \cap \{y : y \leq m\}$. If none of these lead to a mind change, (that is, the conjecture of M remains $M(\sigma)$), then M' repeats its old conjecture $P(\sigma, m, S, t)$. Otherwise, the learner goes to Discovery phase, where initially the conjecture is $Q(m', S, t+1, 0)$.

Discovery phase:

In discovery phase, the learner will try to find all the elements in the input language which are $\leq m'$. This can be done by using conjectures $Q(m', S', t+1, i)$, for $i = 0$ to m'. During this phase, S' starts by being S (see the last conjecture at the end of normal phase). Any new input x which is not in S' is added to S' as well as any $i \leq m'$ which is discovered to be in the input, is added to S'. Once all the elements $\leq m'$ are determined, the learner conjectures $P(\sigma', m', S', t+1)$, where σ' seems like the smallest stabilizing sequence for M on S' as discovered in $t+1$ steps (here by the smallest stabilizing sequence for M on S' discovered in $t+1$ steps we mean the smallest σ' such that [content(σ') $\subseteq S'$ and for all $x \in S'$, $M(\sigma') = M(\sigma' \diamond x)$] or [content($\sigma'$) $= S'$ and $|\sigma'| = t + 1 + card(S')$] — the second condition is used just to make sure that the search terminates, and in the second case contains all the elements seen so far).

We now argue that if M learns the input language, then so does M'. Note that discovery phases are always run for finite time. Also, at the end of discovery phase S' will contain all the inputs that has been seen so far by M' (plus maybe some others). Thus, if there are finitely many phases, then the last phase must be normal, and M' never finds any mind change beyond σ (where σ is as in $P(\sigma, m, S, t)$ output in the final normal phase); thus σ is the stabilizing sequence for M on the input language, and M' learns L.

On the other hand, if the number of phases is infinite, then S keeps containing more and more of L, and t is unbounded (m is also unbounded or becomes the maximal element of the input language), and thus eventually σ must become a stabilizing sequence for M on L, a contradiction to infinitely many phases. □

Our next theorem shows that, for iterative learners (not getting negative counterexamples to conjectures), the number of elements can sometimes give advantage over n feedback queries and access to the maximal element seen so far.

Theorem 25. *There exists a $\mathcal{L}$ such that* *(a) $\mathcal{L}$ can be iteratively learnt when given the number of elements in the input seen so far as additional information (such a learner, however, may not be total).*

(b) For all n, $\mathcal{L}$ cannot be iteratively learnt by an n-feedback learner even if it gets the maximal element as additional information.

Proof. Let $M_0, M_1, \ldots$ denote a recursive enumeration of feedback query learners using maximal elements, where the number of queries used by M_n is at most n.

Using Operator recursion theorem, there exists a recursive 1–1 increasing function p such that $W_{p(n)}$ may be defined as given below. Along with $W_{p(n)}$, we will also try to define $x_{n,0} < x_{n,1} < \ldots$ as well as $\sigma_{n,0} \subseteq \sigma_{n,1} \subseteq \ldots$. Always, $x_{n,s} = 1 + max(\text{content}(\sigma_{n,s}))$. Note that $x_{n,s}$ is defined iff $\sigma_{n,s}$ is defined. $W_{p(n)}$ = union of all content$(\sigma_{n,s})$, where $\sigma_{n,s}$ gets defined.

Initially, $\sigma_{n,0} = \lambda$ and $x_0 = 1$.

Definition of $\sigma_{n,s+1}$:
1. Wait until $M_n(\sigma_{n,s})$ and $M_n(\sigma_{n,s} \diamond \langle n, j, x_s\rangle)$ get defined for each $j \leq 2n+2$.
2. If $M_n(\sigma_{n,s} \diamond \langle n, j, x_s\rangle)\downarrow \neq M_n(\sigma_{n,s})\downarrow$ for some j, Then
    Choose the least such j. Let j', j'' be least such that $j \neq j'$, $j' \neq j''$, $j \neq j''$.
    Let $\sigma_{n,s+1} = \sigma_{n,s} \diamond \langle n, j, x_s\rangle \diamond \langle n, j', x_s\rangle \diamond \langle n, j'', x_s\rangle$.
    Let $x_{s+1} = 1 + max(\text{content}(\sigma_{n,s+1}))$.
   Else Loop forever ($\sigma_{n,s+1}$ does not get defined in this case).

Note that $card(\sigma_{n,s}) = 3s$.

The class $\mathcal{L}$ will consist of the following languages (for each n),
(i) $W_{p(n)}$,
(ii) the languages content$(\sigma_{n,s}) \cup \{\langle n, j, x_s\rangle, \langle n, 2n+3+j, x_s\rangle\}$, for $j \leq 2n+2$, whenever $\sigma_{n,s}$ is defined.
(iii)In case $\sigma_{n,s}$ is defined and $M_n(\sigma_{n,s} \diamond \langle n, j, s\rangle)\downarrow = M_n(\sigma_{n,s})\downarrow$, for all $j \leq 2n+2$, then $\mathcal{L}$ will additionally contain content$(\sigma_{n,s}) \cup \{\langle n, j_1, s\rangle, \langle n, 2n+2, s\rangle\}$ and content$(\sigma_{n,s}) \cup \{\langle n, j_1, s\rangle, \langle n, j_2, s\rangle, \langle n, 2n+2, s\rangle\}$, where
    $j_1 < 2n+2$ is maximal such that M_n on previous conjecture $M(\sigma_{n,s})$ and new input $(\langle n, 2n+2, s\rangle)$ does not query $\langle n, j_1, s\rangle$ and
    $j_2 < j_1$ is maximal such that M_n on previous conjecture ($M(\sigma_{n,s})$ and new input $\langle n, 2n+2, s\rangle$) or $\langle n, j_1, s\rangle$ does not query $\langle n, j_2, s\rangle$ (note that in this case, $\sigma_{n,s+1}$ does not get defined, and $W_{p(n)} = \text{content}(\sigma_{n,s})$).

Claim 26. *$\mathcal{L}$ can be iteratively learnt using the number of elements in the input seen as additional information.*

Proof. An iterative learner (with the number of elements in the input seen as additional information) can learn $\mathcal{L}$ by initially outputting a grammar for $\emptyset$. It can determine $p(n)$ when it first sees $\langle n, j, x\rangle$ for some j, x. Beyond (and including) this point, if it ever sees an element of form

$\langle n, j, x\rangle$, where $j > 2n+2$, it will make the conjecture content$(\sigma_{n,s}) \cup \{\langle n, j, x\rangle, \langle n, j-2n-3, x\rangle\}$ (where s is such that $x = x_s$), as in clause (ii) in the definition of $\mathcal{L}$ and never change its mind thereafter.

Otherwise, if the number of input elements is $3s + 1$, for some s, then the learner outputs $p(n)$. If the number of elements seen is $3s + 2$ or $3s + 3$, then it finds x_s and $\sigma_{n,s}$ (which has $3s$ number of elements). It then waits until $M_n(\sigma_{n,s})$ and $M_n(\sigma_{n,s} \diamond \langle n, j, x_s\rangle)$ get defined for each $j \leq 2n + 2$. If there exists a $j \leq 2n + 2$, such that $M_n(\sigma_{n,s} \diamond \langle n, j, x_s\rangle) \neq M_n(\sigma_{n,s})$, then it continues to output $p(n)$. Otherwise, it determines the maximal j_1 such that M_n on the previous conjecture $M_n(\sigma_{n,s})$ and new input $\langle n, 2n+2, s\rangle)$ does not query $\langle n, j_1, s\rangle$ and maximal $j_2 < j_1$ such that M_n on the previous conjecture $M_n(\sigma_{n,s})$ and new input being $\langle n, 2n + 2, s\rangle$ or $\langle n, j_1, s\rangle)$ does not query $\langle n, j_2, s\rangle$. The learner then outputs a grammar for content$(\sigma_{n,s}) \cup \{\langle n, j_1, x_s\rangle, \langle n, j_2, x_s\rangle, \langle n, 2n+2, x_s\rangle\}$, if the number of elements was $3s+3$ and outputs a grammar for content$(\sigma_{n,s}) \cup \{\langle n, j_1, x_s\rangle, \langle n, 2n + 2, x_s\rangle$, if the number of elements is $3s + 2$. It is easy to verify that the above learner will iteratively learn $\mathcal{L}$ using the number of elements as input. This completes the proof of the claim.

Claim 27. *M_n cannot iteratively learn $\mathcal{L}$ using n-feedback queries, even if it is given the maximal element in the input so far as additional information.*

Proof. To see this, suppose infinitely many $\sigma_{n,s}$ get defined. Then clearly M_n does not **TxtIt** learn $W_{p(n)}$ using n-feedback from the text $\bigcup_s \sigma_{n,s}$, as there are infinitely many mind changes by M_n on the text.

On the other hand, if $\sigma_{n,s+1}$ does not get defined, then if step 1 does not finish, then M_n does not learn the language content$(\sigma_{n,s}) \cup \{\langle n, j, x_s\rangle, \langle n, 2n + 3 + j, x_s\rangle\}$ for the j for which $M_n(\sigma_{n,s} \diamond \langle n, j, x_s\rangle)$ does not converge.

If step 2 does not finish, then $M_n(\sigma_{n,s} \diamond \langle n, j, x_s\rangle) = M_n(\sigma_{n,s})$, for each $j \leq 2n + 2$. Let $j_1 < 2n + 2$ be maximal such that M_n on the previous conjecture $M_n(\sigma_{n,s})$ and new input $\langle n, 2n + 2, s\rangle$ does not query $\langle n, j_1, s\rangle$ and let $j_2 < j_1$ be maximal such that M_n on the previous conjecture $M_n(\sigma_{n,s})$ and new input being $\langle n, 2n+2, s\rangle)$ or $\langle n, j_1, s\rangle$ does not query $\langle n, j_2, s\rangle$. Then, M_n fails to identify at least one of content$(\sigma_{n,s}) \cup \{\langle n, j_1, s\rangle, \langle n, 2n + 2, s\rangle\}$ and content$(\sigma_{n,s}) \cup \{\langle n, j_1, s\rangle, \langle n, j_2, s\rangle, \langle n, 2n+2, s\rangle\}$, as M_n converges to the same grammar on both $\sigma_{n,s} \diamond \langle n, j_1, s\rangle \diamond \langle n, 2n + 2, s\rangle^\infty$ and $\sigma_{n,s} \diamond \langle n, j_2, s\rangle \diamond \langle n, j_1, s\rangle \diamond \langle n, 2n + 2, s\rangle^\infty$. □

A similar idea can be used to show that, for iterative learners, the number of elements can give more advantage than n stored elements seen on the input.

Theorem 28. *There exists a $\mathcal{L}$ such that*

(a) $\mathcal{L}$ can be iteratively learnt when given the number of elements in the input seen so far as additional information (such a learner however may not be total).

(b) For all n, $\mathcal{L}$ cannot be iteratively learnt by an n-memory bounded learner.

6 Using Length of Input as Additional Information

The length of the input seen so far can potentially be viewed as an alternative to the number of elements seen so far as a source of additional information for **NCIt**-learners. However, we show in this section that, for **NCIt**-learners, it can be replaced by access to the maximal element.

Theorem 29. *Suppose $\mathcal{L}$ is* **NCIt***-learnable using the length of input as additional information. Then $\mathcal{L}$ is* **NCIt***-learnable using the maximal element as additional information.*

Proof. Suppose M is **NCIt** learner which uses length of input. We will define M' below which uses the maximal element and **NCIt**-learns the languages which are **NCIt**-learnt by M.

Below ψ will denote a finite partial function such that $\psi(i)$ is a counterexample (or #) to conjecture i. The defined part of ψ will always be correct with respect to the input language L.

Let P be a 1–1 recursive function such that $W_{P(\sigma,\psi)} = W_{M(\sigma)}$, if $M(\sigma) = M(\sigma\#^k)$ for all k (where counterexamples are provided according to ψ, and $\psi(M(\sigma'))$ is defined for all prefixes σ' of σ); Otherwise $W_{P(\sigma,\psi)} = \mathbb{N}$ (this condition applies if either $\psi(M(\sigma'))$ is not defined for some prefix σ' of σ, or $M(\sigma) \neq M(\sigma\#^k)$ for some k). Here we assume without loss of generality that $P(\sigma,\psi) = P(\sigma,\psi')$, as long as ψ,ψ' coincide on $M(\sigma')$ for each prefix σ' of σ (this is for ease of notation).

Initially, M' outputs a conjecture for $\mathbb{N}$ (which we assume without loss of generality not to be syntactically same as any of the other conjectures used below). If there is no counterexample, then we are done. Otherwise M' goes to stage 0 below. Note that M' can remember ψ, stage number, phase and which part of the phase (in case of phase 1) it is in, by just padding its conjecture appropriately; Thus, for phase 1, we essentially describe it as if M' can remember all the data it has seen since the phase started — in phase 1, we are not concerned about converging to a hypothesis, but just about finding certain σ_s — thus all new input seen can be padded.

Stage s

Phase 1:

1. Suppose m is the maximal element seen so far. Below τ will denote the sequence of elements seen in the input since this phase started (note that this τ will keep getting updated with time, based on which step we are executing).
2. Let Y denote the set of elements $\leq m$ which belong to the input language (this can be determined using (padded) conjectures for $\{x\}$, with $x \leq m$). Note here that each such conjecture would update τ too.
3. Loop for $t = 0$ to ∞:
   - 3.1 Update ψ by finding the value of ψ on the least number on which it is not defined. This can be done by conjecturing the least input on which ψ is not defined (τ correspondingly gets updated).

     Suppose $t = \langle t', t''\rangle$. Let σ_s be the sequence with canonical code t'.

     Below, in simulation of M counterexamples are provided using ψ.

     If

3.2 content(σ_s) $\subseteq Y \cup$ content(τ)), $\psi(M(\sigma'))$ is defined for each prefix σ' of σ_s and

3.3 $\psi(P(\sigma_s, \psi)) = \#$, and

3.4 $M(\sigma_s \#^x \cdot x) = M(\sigma_s)$, for all $x \in Y \cup$ content(τ).

Then go to step 4. Otherwise go to next iteration of the loop.

End Loop

4. If appropriate σ_s is found, then M' outputs (padded) $P(\sigma_s, \psi)$ and goes to Phase 2.

Phase 2

On new input x check if $M(\sigma_s) = M(\sigma_s \#^x x)$. If not, then go to stage $s+1$. Otherwise repeat $P(\sigma_s, \psi)$ as the conjecture.

End stage s

Now we claim that above M' **NCIt**-learns (using the maximal element seen as additional information) any language which is **NCIt**-learnt by M (using the length of input as additional information).

Let L be the input language. Values of variables below are as at the corresponding stage/phase.

If the input language is $\mathbb{N}$, then clearly M' learns it.

Now suppose the above learner gets stuck in phase 2 of some stage s. Let T' be a text for L where $T'(x) = x$, if $x \in L$, and $T'(x) = \#$ otherwise. Then clearly, for all n, $M(\sigma_s T'[n]) = M(\sigma_s)$ (otherwise either $P(\sigma_s, \psi)$ would have generated a counterexample, or a mind change would have been found on $M(\sigma_s \#^x x)$ for some $x \in Y \cup$ content(τ) as at step 3.4 of phase 1, or when input x is received in phase 2; here note that Y contains all the data seen in stages before stage s and ($M(\sigma_s) = M(\sigma_s \#^x)$ and $M(\sigma_s) = M(\sigma_s T'[x])$) implies $M(\sigma_s \#^x x) = M(\sigma_s T'[x+1])$). Moreover, all the answers given by ψ are correct. Thus, M' **NCIt**-learns L as M does so.

If M' gets stuck at phase 1, then there is no stabilizing sequence for M on input L, and thus M does not **NCIt**-learn L. (Here note that Y contains all the data seen by M' in stages before stage s, and thus all elements of the input language are eventually in $Y \cup$ content(τ), as step 3.1 gets executed infinitely often).

We now argue that there cannot be infinitely many stages. Let $\langle t', t'' \rangle$ be least such that σ with canonical index t' satisfies: (i) content(σ) $\subseteq L$, (ii) for all $x \in L$, $M(\sigma \#^x x) = M(\sigma)$ and $M(\sigma \#^k) = M(\sigma)$ for all k. Note that there exists such a σ as every stabilizing sequence satisfies these properties.

Let s be large enough so that (i) input text $T[s]$ contains all elements in content(σ), (ii) for all $\langle t'_1, t''_1 \rangle < \langle t', t'' \rangle$, for γ with canonical index t'_1, either content(γ) $\not\subseteq$ content(T), or $M(\gamma) \neq M(\gamma \#^x x)$ for some $x \in$ content($T[s]$), or $M(\gamma \#^k) \neq M(\gamma)$, for some k. (Here, note that after the first execution of step 3.1 of the beginning of stage s at least the first s elements of the input text are already in Y.) Now, in the loop at stage s, Phase 1, σ would be chosen as σ_s, and thus the learner will not leave stage s anymore. □

7 Robustness of NCIt-learning with Additional Information

In this section we briefly consider what happens if instead of giving the maximal/number of elements as additional information the learner is only given an upper bound on these values or an approximate value which is within an additive constant c of the actual value.

Note that an upper bound on the maximal element implies an upper bound on the number of elements (which means that a bound on the number of elements seen can always be simulated using a bound on the maximal element).

First we note that the maximal element can be replaced by an upper bound in the context of Theorem 19.

Now, the next question is whether Theorem 20 that shows the strength of **NCIt**-learners using the maximal element against the ones using the number of elements can be extended to the case when only an upper bound for the maximal element is available to the learner. We don't know the answer to this question.

Theorem 22 shows the strength of the maximal element (or 1-cell memory) and the number of elements available to **NCIt**-learners together. The proof of this theorem does not work if, on the positive side, the learner is given an upper bound on the number of elements, or an approximation to the number of elements within an additive constant. However, one can modify the proof of Theorem 22 to work for the case when the positive side is given an approximation to the upper bound on the number of elements (within an additive constant c), by replacing $L \in \mathcal{L}$ to L', where

$x \in L$ iff L' contains $(6c+3)(x-1)+2c+2+j$, for $0 \leq j \leq 2c+1$.

The idea here is that each x is mapped to a group of $6c+3$ elements, where the least and the highest $2c+1$ elements are not in the language L', and the middle $2c+1$ ones are in L' iff x is in L; this essentially allows a learner, given an approximation within a "constant c" for the number of elements in the language (in case it is finite), to compute the actual number of elements in the input language. This is enough for learning the class $\mathcal{L}$ as in Theorem 22. The diagonalization proof as in Theorem 22 also can be adjusted appropriately, as the proof there worked for all possible n-feedback and n-memory learners. A similar idea can be used for similar cases below.

Also note that Theorem 22 does not work if the upper bound on the number of elements is given on the positive side, as the maximal element bounds the number of elements too. So the diagonalization against n-memory bounded **NCIt**-learner does not work in this case.

For Theorem 21, the proof does not work if we are given an upper bound on the number of elements. However, the proof can be modified to work for the case when an upper bound on the number of elements is given to the learner. This is done by first partitioning $\mathbb{N}$ into blocks $I_0, I_1, I_2, \ldots$, where I_k is of size $k+1$. The languages in $\mathcal{L}_2$, are of the form $\{\langle e, x\rangle : x \in I_k, k \in D\} \cup \{\langle 0, y\rangle\}$, for some finite D. Thus, essentially, the upper bound on the number of elements gives away the upper bound on the maximal for languages in $\mathcal{L}_2$. This allows one to identify the class $\mathcal{L}$. The diagonalization proof can easily be modified to use the updated class.

Theorem 24 shows that, for **NCIt**-learners converging on all inputs, the number of elements can be simulated by the maximal element. This simulation holds if only an upper bound on the maximal element is given.

For Theorem 25,showing advantage of iterative learners using access to the number of elements over the ones using n-feedback and the maximal element, the proof can be modified to handle the case when, on the positive side, the learner is given an approximation to the number of elements within an additive constant c (by using $2c+1$ cylindrification). The proof, of course, cannot be modified to work for an upper bound on the number of elements, as it is bounded by the upper bound on the maximal element.

8 Conclusions

As we have shown, additional information of the types studied in this paper can add interesting new capabilities to iterative learners getting negative examples to conjectures containing data in excess of the target language. Some problems related to comparisons of help provided by additional information remain open (they are mentioned in Section 5), and solving these problems can offer new (and, possibly, unexpected) insight into advantages of using additional information of certain types for the learners in question. Influence of noise on additional information has been discussed in Section 7, however, many questions remain here open as well. Similarly to [JK07], one might also consider different types of negative examples (refuting conjectures containing extra elements) by iterative learners and explore how these different types of negative examples may interplay with different types of additional information. Yet another interesting area of research is studying iterative learnability with counterexamples and additional information of specific indexed classes of languages (for example, regular languages or patterns) — as we have shown all such classes are learnable class-preservingly, and, therefore, one can now study if and when learnability of such classes may be efficient.

A general open problem for iterative learners of any type using additional (bounded) memory is whether a *multiset* type memory (when a learner can store the same inputted item several times) can have advantage over a *set* type memory (where every item is stored just once). The first insight is that no such advantage could be possible — however, we have not be able to find an answer to this very interesting problem.

References

[Ang88] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

[BA96] R. Brachman and T. Anand. The process of knowledge discovery in databases: A human centered approach. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusam, editors, *Advances in Knowledge Discovery and Data Mining*, pages 37–58. AAAI Press, 1996.

[BB75] L. Blum and M. Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28:125–155, 1975.

[Blu67] M. Blum. A machine-independent theory of the complexity of recursive functions. *Journal of the ACM*, 14:322–336, 1967.

[CJLZ99] J. Case, S. Jain, S. Lange, and T. Zeugmann. Incremental concept learning for bounded data mining. *Information and Computation*, 152(1):74–110, July 1999.

[CL82] J. Case and C. Lynes. Machine inductive inference and language identification. In M. Nielsen and E. M. Schmidt, editors, *Proceedings of the 9th International Colloquium on Automata, Languages and Programming*, volume 140 of *Lecture Notes in Computer Science*, pages 107–115. Springer-Verlag, 1982.

[FPSS96] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusam, editors, *Advances in Knowledge Discovery and Data Mining*, pages 1–34. AAAI Press, 1996.

[Ful90] M. Fulk. Prudence and other conditions on formal language learning. *Information and Computation*, 85:1–11, 1990.

[Gol67] E. M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[HU79] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

[JK07] S. Jain and E. Kinber. Iterative learning from positive data and negative counterexamples. *Information and Computation*, 205(12):1777–1805, 2007.

[JK08] S. Jain and E. Kinber. Learning languages from positive data and negative counterexamples. *Journal of Computer and System Sciences*, 74(4):431–456, 2008. Speical Issue: Carl Smith memorial issue.

[LZ92] S. Lange and T. Zeugmann. Types of monotonic language learning and their characterization. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 377–390. ACM Press, 1992.

[LZ96] S. Lange and T. Zeugmann. Incremental learning from positive data. *Journal of Computer and System Sciences*, 53:88–103, 1996.

[LZ06] Y. Li and W. Zhang. Simplify support vector machines by iterative learning. *Neural Processsing Information - Letters and Reviews*, 10:11–17, 2006.

[OSW86] D. Osherson, M. Stob, and S. Weinstein. *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists*. MIT Press, 1986.

[Pop68] K. Popper. *The Logic of Scientific Discovery*. Harper Torch Books, New York, second edition, 1968.

[Rog67] H. Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, 1967. Reprinted by MIT Press in 1987.

[Wie76] R. Wiehagen. Limes-Erkennung rekursiver Funktionen durch spezielle Strategien. *Journal of Information Processing and Cybernetics (EIK)*, 12:93–99, 1976.

[ZL95] T. Zeugmann and S. Lange. A guided tour across the boundaries of learning recursive languages. In K. Jantke and S. Lange, editors, *Algorithmic Learning for Knowledge-Based Systems*, volume 961 of *Lecture Notes in Artificial Intelligence*, pages 190–258. Springer-Verlag, 1995.