Mind Change Speed-up for Learning Languages from Positive Data

Sanjay Jain [a,1] and Efim Kinber [b]

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

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

Abstract

Within the frameworks of learning in the limit of indexed classes of recursive languages from positive data and automatic learning in the limit of indexed classes of regular languages (with automatically computable sets of indices), we study the problem of minimizing the maximum number of mind changes $\mathbf{F_M}(n)$ by a learner $\mathbf{M}$ on all languages with indices not exceeding n. For inductive inference of recursive languages, we establish two conditions under which $\mathbf{F_M}(n)$ can be made smaller than any recursive unbounded non-decreasing function. We also establish how $\mathbf{F_M}(n)$ is affected if at least one of these two conditions does not hold. In the case of automatic learning, some partial results addressing speeding up the function $\mathbf{F_M}(n)$ are obtained.

Keywords: Inductive Inference, Algorithmic and automatic learning, mind changes, speedup.

1 Introduction

In this paper, we consider a popular model for learning languages in the limit from infinite positive data (inductive inference), as defined by M. Gold in [Gol67] (in the sequel, we refer to it as **TxtEx**): a learner is an algorithmic device that, given access to potentially all positive data (as a stream of data items, intermittent with a special character representing “no data at this moment”), produces a (potentially infinite) sequence of conjectures, and eventually stabilizes on a correct grammar for the target language. Specifically, we

[1] Supported in part by NUS grant number C-252-000-087-001.

concentrate on learnability of so-called *indexed* classes of languages — represented by computable numberings of languages with uniformly decidable membership problem; these classes represent practically interesting families of languages, in particular, the class of regular languages as represented by all finite automata or regular expressions and its practically important subclasses, and the class of pattern languages represented by patterns [Ang80].

There are many different measures of complexity for learning languages in the limit [CL82,CS83,DS86,Wie86,JS97,FKS95]. One obvious natural measure of complexity is the number of mind changes that a learner makes on a target language before stabilizing on a correct grammar for it. As there are infinitely many languages in the target class, it is natural to consider the maximum number of mind changes that a learner $\mathbf{M}$ makes on the first $n+1$ languages in the numbering defining the target class; in the sequel, we denote this number by $\mathbf{F}_{\mathbf{M}}(n)$ (another approach to mind change complexity was suggested in [LZ93b]). This measure of complexity of inductive inference, in the context of learning indexed families of recursive functions, was first suggested by J. Bārzdiņš and R. Freivalds in [BF72], where they also initiated a study of the bounds on the function $\mathbf{F}_{\mathbf{M}}(n)$. It is easy to see that $\mathbf{F}_{\mathbf{M}}(n)$ can be bounded by n — the learner can use the so-called "identification by enumeration" strategy, whereby all functions in the numbering consistent with the input data seen so far are tried, starting from the first one, until a (smallest) index of the target function is found. However, Bārzdiņš and Freivalds showed in [BF72] (providing full proof in [BF74]) that the linear upper bound on $\mathbf{F}_{\mathbf{M}}(n)$ can be reduced exponentially — to $\log n + \log\log n + o(\log\log n)$, if the learner is allowed to use programs of a general type (from a universal acceptable numbering of all programs) rather than indices in the numbering of the target class. They also established a nearly matching lower bound for the function $\mathbf{F}_{\mathbf{M}}(n)$ (having shown that there exists an indexed class of functions where no strategy can use less than nearly $\log n$ mind changes). In the paper [Bār74b], J. Bārzdiņš showed that the lower bound on the number of mind changes jumps to nearly n, if the numbering defining the target class of functions is used as the hypotheses space.

In the paper [BKP74], the authors studied the following problem: is it possible to "speed up" learning of indexed classes of functions achieving as slow growth of the function $\mathbf{F}_{\mathbf{M}}(n)$ as possible? More specifically, if and when is it possible, given any total recursive function $r(n)$ and any learner $\mathbf{M}$ for an indexed class $\mathcal{L}$, to find another learner $\mathbf{M}'$ such that, for all n, $r(\mathbf{F}_{\mathbf{M}'}(n)) \leq \max(\{\mathbf{F}_{\mathbf{M}}(n), c\})$, for some constant c? They suggested to call such a provable statement for a class $\mathcal{L}$ "absolute speed-up theorem" (AST, for brevity), and established validity of AST for any class $\mathcal{L}$ of recursive functions with decidable equivalence problem and not learnable with a constant number of mind changes.

In this paper, we study possibilities of mind change speed-ups in two different contexts. First, we consider **TxtEx**-learning (from all positive data) of indexed families of recursive languages. Secondly, we consider learning in the limit, from positive data, *automatic* classes of languages by *automatic* learners; such an indexed class of languages is defined by a finite automaton (the study of inductive inference in this context was initiated in [JLS10]).

In the general case of **TxtEx**-learning of indexed families, we establish the conditions under which AST is possible: we show that AST holds if (a) the equivalence problem for the languages in the class is decidable and (b) inclusion of one language in another one implies their equality (Theorem 4). Note that the condition (a) typically holds for practically important indexed families of languages (for example, the class of regular languages indexed by finite automata and the class of pattern languages indexed by patterns). In light of this, the condition (b) is really the important criterion deciding if the AST can work for an indexed class. This condition is quite simple and can be typically tested for many practically useful indexed classes. Then, we show that, if the condition (a) holds and the condition (b) does not (and yet there are no subset chains of languages of length more than 2), then $\mathbf{F_M}(n)$ can grow faster than any fixed recursive function (Theorem 9). We also consider the case when the condition (b) holds, but (a) does not. It turns out, that, in this case, any class can be learned with $O(\log n)$ upper bound on $\mathbf{F_M}(n)$ (Theorem 12), and there exists a class with the lower bound of $\log n - o(\log n)$ on $\mathbf{F_M}(n)$ for any learner $\mathbf{M}$ (Theorem 11). As in the case of learning functions, the algorithm, providing logarithmic upper bound on $\mathbf{F_M}(n)$ utilizes a “majority vote” strategy.

Interestingly, if a learner witnessing AST is required to conjecture grammars only for the languages in the class, then it cannot be made consistent with the input seen so far: for such consistent learners, we show that, for some classes, the lower bound on $\mathbf{F_M}(n)$ is $\log n + 1$ (cf. Theorem 8).

For the automatic case, the definition of $\mathbf{F_M}$ needs to be readjusted, as indices of languages are strings, and the set of indices must be regular; in addition, we require learners to be computable by finite automata (automatic). Accordingly, we consider a (natural) ordering of all indices and define $\mathbf{F_M}(w)$ as the maximum of the number of mind changes on all languages with indices not length-lexicographically greater than w with respect to the given ordering.

We have not been able to find a reasonable range of automatic classes for which AST holds. Yet, we obtained some interesting partial results. First, we show that, for any non-decreasing unbounded automatic function, there is an automatic class that can be learned by an automatic learner with $\mathbf{F_M}(w)$ not exceeding this function; yet AST is not possible for this class, as the function provides also the matching lower bound on $\mathbf{F_M}(w)$ (Theorem 14). Then we show that, for a range of automatic classes satisfying a simple condition, $\mathbf{F_M}(w)$ can be made smaller than any unbounded non-decreasing recursive function if an automatic learner uses *fat* texts, where every input datum appears infinitely many times (Theorem 16). This result works also for automatic learners using arbitrary input texts if the languages in the class satisfy the additional condition of being pairwise infinitely different. Although, our results in this section do not directly deal with a more practically interesting AST problem for learning automatic classes with no strong restrictions on either stream of input data or the class to be learnt, they certainly shed light on the difficulties of solving the AST problem without these restrictions.

Mind changes have played an important role in other fields besides inductive inference, such as in computational complexity to determine the powers of Boolean Hierarchy, query order, etc. [KSW87,CGH$^+$88,Bei91,HHW98].

2 Preliminaries

Let N denote the set of natural numbers. A language is a subset of N. The symbol $\emptyset$ denotes the empty set. Symbols $\subseteq, \supseteq, \subset, \supset$, respectively, denote subset, superset, proper subset and proper superset. Furthermore, $\max(S), \min(S)$ and $\mathrm{card}(S)$, respectively, denote the maximum, minimum and cardinality of a set S, where $\max(\emptyset) = 0$ and $\min(\emptyset) = \infty$. We use $\mathrm{card}(S) \leq *$ to denote that the cardinality of S is finite. When we say that two languages L and L' are incomparable, then we mean that $L \not\subseteq L'$ and $L' \not\subseteq L$.

We let $\langle \cdot, \cdot \rangle$ stand for an arbitrary, computable, one-to-one encoding of all pairs of natural numbers onto N [Rog67]. The functions π_1 and π_2 denote the corresponding projection functions. Similarly, one can define $\langle \cdot, \cdot, \ldots, \cdot \rangle$ coding multiple arguments, and the corresponding projection functions. We assume these pairing functions to be monotonically increasing in all their arguments.

We let φ denote a fixed *acceptable* programming system for the partial computable functions [Rog67]. The i-th partial computable function in the system φ is denoted by φ_i. The set of all recursive functions is denoted by $\mathcal{R}$. When considering partial computable functions with multiple arguments, we assume that the inputs are coded using the pairing function described above. Φ denotes a fixed Blum complexity measure [Blu67] for the φ system. φ_i^s denotes the function: $\varphi_i^s(x) = \varphi_i(x)$, if $x < s$ and $\Phi_i(x) < s$; otherwise $\varphi_i^s(x)$ is undefined. We let $W_i = \mathrm{domain}(\varphi_i)$ and $W_i^s = \mathrm{domain}(\varphi_i^s)$.

For a language L, $L[m] = \{x \leq m : x \in L\}$. For a function h, $h^{-1}(m)$ denotes the set of n such that $h(n) = m$.

A finite sequence σ is a mapping from an initial segment of N into $(N \cup \{\#\})$. We let Λ denote the empty sequence. The content of σ, denoted content(σ), is the set of natural numbers in the range of σ. The length of σ, denoted $|\sigma|$, is the number of elements in the domain of σ. SEQ denotes the set of all finite sequences. A text T is a mapping from N to $(N \cup \{\#\})$. The content of T, denoted content(T), is the set of natural numbers in the range of T. A text T is for a language L iff content$(T) = L$. $T[n]$ denotes the initial segment of T of length n, and $\sigma[n]$ denotes the initial segment of σ of length n. Intuitively, #'s denote pauses in the presentation of data. A text T is called *fat* [OSW86] if for every $x \in$ content(T), there exist infinitely many n such that $T(n) = x$.

A language learning machine is an algorithmic mapping from SEQ to $N \cup \{?\}$. Intuitively, ? denotes that the learner does not have enough data to form a conjecture. We let $\mathbf{M}$, with or without decorations, range over learning machines. If, for all but finitely many n, $\mathbf{M}(T[n]) = i$, then we say that $\mathbf{M}(T)\downarrow = i$ (or simply, $\mathbf{M}(T) = i$). If there exists an i such that $\mathbf{M}(T)\downarrow = i$, then we say that $\mathbf{M}(T)$ converges (written: $\mathbf{M}(T)\downarrow$); otherwise, we say that $\mathbf{M}(T)$ diverges or $\mathbf{M}(T)$ is undefined (written: $\mathbf{M}(T)\uparrow$).

Definition 1 [Gol67] (a) $\mathbf{M}$ **TxtEx**-identifies a language L (written: $L \in \mathbf{TxtEx}(\mathbf{M})$) iff for all texts T for L, $\mathbf{M}(T)\downarrow$ and $W_{\mathbf{M}(T)} = L$.

(b) $\mathbf{M}$ **TxtEx**-identifies a class $\mathcal{L}$ of languages iff $\mathbf{M}$ **TxtEx**-identifies each $L \in \mathcal{L}$.

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

For a learner $\mathbf{M}$, a text T, and $n \in N$, we let $\mathrm{MC}_{\mathbf{M}}(T[n])$ denote the number of mind changes [CS83,CL82] made by $\mathbf{M}$ on $T[n]$, that is, card$(\{r < n : ? \neq \mathbf{M}(T[r]) \neq \mathbf{M}(T[r+1])\})$. Similarly, $\mathrm{MC}_{\mathbf{M}}(T)$ denotes the number of mind changes [CS83,CL82] made by $\mathbf{M}$ on T, that is, card$(\{r :? \neq \mathbf{M}(T[r]) \neq \mathbf{M}(T[r+1])\})$. We let $\mathrm{MC}_{\mathbf{M}}(L)$ denote the maximum over $\mathrm{MC}_{\mathbf{M}}(T)$ for all texts T for L. One can assume without loss of generality that, if $\mathbf{M}(\sigma) \neq ?$ and $\sigma \subseteq \tau$, then $\mathbf{M}(\tau) \neq ?$.

A learner $\mathbf{M}$ is said to be *consistent* [Ang80,Bār74a] if for all $\sigma \in$ SEQ, content$(\sigma) \subseteq W_{\mathbf{M}(\sigma)}$.

An indexed family is a family $\mathcal{L} = (L_i)_{i \in N}$ of languages such that, $\{(i, x) : x \in L_i\}$ is recursive. When dealing with indexed families, we let $\mathbf{F}_{\mathbf{M}}(i) =$ maximum over $\mathrm{MC}_{\mathbf{M}}(T)$ on any input text T for a language L_j, $j \leq i$.

Often, when learning indexed families, instead of using the acceptable programming system $W_0, W_1, \ldots$ as hypothesis space, we use an indexed family, $(H_i)_{i \in N}$, as hypothesis space. That is, in Definition 1(a), we require $H_{\mathbf{M}(T)} = L$, instead of requiring $W_{\mathbf{M}(T)} = L$. This model of learning is said to be *class preserving* [LZ93a,ZZ08] if $\{H_i : i \in N\} = \{L_i : i \in N\}$. In theorems in the sequel, for positive learnability statements, by default, we take the hypothesis space $H_i = L_i$, unless specified otherwise. For non-learnability statements, we allow acceptable programming system $(W_i)_{i \in N}$ as hypothesis space, (and thus the diagonalization works against arbitrary hypothesis spaces).

We now formally define AST.

Definition 2 Suppose an indexed family $\mathcal{L} = (L_i)_{i \in N}$ is given. We say that $\mathcal{L}$ satisfies *absolute speed-up theorem* (AST) if for any recursive function $r(\cdot)$ and a learner $\mathbf{M}$ for $\mathcal{L}$, there exists another learner $\mathbf{M}'$ and a constant c such that, for all n, $r(\mathbf{F}_{\mathbf{M}'}(n)) \leq \max(\{\mathbf{F}_{\mathbf{M}}(n), c\})$.

3 Mind Change Speed-up for Learning Recursive Languages

Our main goal in this section is to establish conditions under which AST holds for learning an indexed class of languages.

First, we note that AST does not hold for some indexed classes. As our first theorem shows, some classes, for example, require linear number of mind changes.

Theorem 3 *Let $L_i = \{j : j < i\}$. Let $\mathcal{L} = \{L_i : i \in N\}$. Then,*

(a) $\mathcal{L}$ can be **TxtEx***-learnt by a learner* $\mathbf{M}$*, such that* $\mathbf{F}_{\mathbf{M}}(n) = n$.

(b) Any learner $\mathbf{M}$ *which* **TxtEx***-learns* $\mathcal{L}$ *has* $\mathbf{F}_{\mathbf{M}}(n) \geq n$.

Proof. (a) Consider a learner $\mathbf{M}$ such that $\mathbf{M}(T[n]) = 0$, if $\text{content}(T[n]) = \emptyset$; otherwise $\mathbf{M}(T[n]) = \max(\text{content}(T[n])) + 1$. Clearly, $\mathbf{M}$ witnesses part (a) (where the hypothesis space used by the learner $\mathbf{M}$ is $(L_i)_{i \in N}$).

(b) For any **TxtEx**-learner $\mathbf{M}$ for $\mathcal{L}$, one can construct σ_i, $i \in N$, such that $\sigma_i \subseteq \sigma_{i+1}$, $\text{content}(\sigma_i) = L_i$, and $\mathbf{M}(\sigma_i)$ is a grammar for L_i. Then $\text{MC}_{\mathbf{M}}(\sigma_i) \geq i$. ∎

Note that for any **TxtEx**-learner $\mathbf{M}$ for an indexed family $\mathcal{L}$, for which number of mind changes cannot be bounded by a constant, one can effectively construct a recursive non-decreasing unbounded function h such that $h(n) \leq \mathbf{F}_{\mathbf{M}}(n)$. Thus, the following theorem gives conditions for AST holding for an indexed class (the actual AST is stated in the corollary).

Proof of the following theorem essentially uses the idea of delaying mind change until it is safe, that is, until all grammars, except for at most one grammar, upto a sufficiently large bound are found to be incompatible with the input data.

Theorem 4 *Suppose $\mathcal{L} = (L_i)_{i \in N}$ is an indexed family for which the equivalence problem is decidable. Furthermore, assume that $L_i \subseteq L_j$ implies $L_i = L_j$.*

Suppose h is a monotonically non-decreasing recursive function, with $\mathrm{range}(h)$ *being unbounded.*

Then, there exists a learner $\mathbf{M}$ which **TxtEx**-*learns $\mathcal{L}$ such that $\mathbf{F_M}(n) \leq h(n)$.*

Proof. The hypothesis space used by the learner $\mathbf{M}$ is $(L_i)_{i \in N}$. Let $H(k) = \min(\{k' : h(k') > k\})$. Let $\mathbf{M}(\Lambda) =?$. Inductively, define $\mathbf{M}(T[n+1])$ as follows.

If for all $j \leq n$, $\mathrm{content}(T[n+1]) \not\subseteq L_j$, then let $\mathbf{M}(T[n+1]) = \mathbf{M}(T[n])$.

Otherwise, let j be least such that $\mathrm{content}(T[n+1]) \subseteq L_j$. If there exists a $j' < H(\mathrm{MC}_{\mathbf{M}}(T[n])+1)$, such that $L_j \neq L_{j'}$ (this can be tested, as the equivalence problem is decidable) and $\mathrm{content}(T[n+1]) \subseteq L_{j'}$, then let $\mathbf{M}(T[n+1]) = \mathbf{M}(T[n])$ (the learner $\mathbf{M}$ "does not want" to change mind to j, as there is a different language containing the same initial segment of input data not "too far" from j — as defined by the function H); otherwise let $\mathbf{M}(T[n+1]) = j$.

Note that if $\mathbf{M}(T[n+1]) = j$, then for all $j' < H(\mathrm{MC}_{\mathbf{M}}(T[n])+1)$, $L_j = L_{j'}$ or $\mathrm{content}(T[n+1]) \not\subseteq L_{j'}$. That is, for all j' such that $h(j') \leq \mathrm{MC}_{\mathbf{M}}(T[n]) + 1$, $L_j = L_{j'}$ or $\mathrm{content}(T[n+1]) \not\subseteq L_{j'}$. Thus, if $\mathrm{content}(T[n+1]) \subseteq L_i$ for an L_i different from L_j, i must be so large that $\mathrm{MC}_{\mathbf{M}}(T[n+1]) < h(i)$. It follows that, given any L_i, for any text T for L_i, $\mathrm{MC}_{\mathbf{M}}(T) \leq h(i)$. Furthermore, $\mathbf{M}$ **TxtEx**-identifies L_i on a text T for L_i, as after it has received $T[n+1]$ such that $\mathrm{content}(T[n+1]) \not\subseteq L_{j'}$ for any j' such that $h(j') \leq \max(\{h(i), 1\})$, we will have $\mathbf{M}(T[n+1]) = i$. ∎

Corollary 5 *Suppose $\mathcal{L} = (L_i)_{i \in N}$ is an indexed family for which the equivalence problem is decidable. Furthermore, assume that $L_i \subseteq L_j$ implies $L_i = L_j$. Then AST holds for $\mathcal{L}$.*

Proof. Suppose $\mathbf{M}$ **TxtEx**-identifies $\mathcal{L}$ and $\mathbf{F_M}$ is the corresponding mind change complexity function. The corollary is trivial if $\mathbf{F_M}$ is bounded by a constant. So assume $\mathbf{F_M}$ is unbounded. Given a recursive function r, define the recursive function h such that, $h(0) = 0$, and $h(n+1) = h(n) + 1$ if $r(h(n)+1) \leq \mathbf{F_M}(n+1)$ as can be verified by running $\mathbf{M}$ on some σ of length at most n, such that content$(\sigma) \subseteq \{x : x \leq n\} \cap L_i$, for some $i \leq n+1$; $h(n+1) = h(n)$ otherwise. Thus, $r(h(n)) \leq \mathbf{F_M}(n)$, for $n \geq 1$. Now the corollary follows from Theorem 4. ∎

The above corollary immediately gives the result of Bārzdiņš, Kinber and Podnieks [BKP74] that, in the case of inductive inference of indexed classes of recursive functions, decidability of the equivalence problem for the functions in an indexed class suffices for AST.

Remark 6 *The conditions of Theorem 4 are not necessary. For example, one can easily transform any indexed class $\mathcal{L} = (L_i)_{i \in N}$ satisfying the conditions of Theorem 4 into a class $\mathcal{L}' = (L'_j)_{j \in N}$ with undecidable equivalence problem and AST holding for it. For this, one takes either $L'_{2j} = L'_{2j+1} = \{2x : x \in L_j\}$ or $L'_{2j} = \{2x : x \in L_j\} \cup \{2 * \langle 2j, r_j \rangle + 1\}$ and $L'_{2j+1} = \{2x : x \in L_j\} \cup \{2 * \langle 2j+1, r_j \rangle + 1\}$, for some appropriate large enough r_j, such that the i-th Turing Machine does not correctly decide whether $L'_{2j} = L'_{2j+1}$.*

Remark 7 *The learner in the proof of Theorem 4 can be made consistent (for indexed families), if the learner is allowed to output N as a conjecture. For this, if the conjecture of the learner in the proof of Theorem 4 is inconsistent (including for the initial conjecture ?), then it is replaced by a conjecture for N. This potentially doubles the number of mind changes made, however this problem can be easily addressed by replacing $h(i)$ by $\lfloor (h(i) \dot{-} 1)/2 \rfloor$ in the construction as in Theorem 4. Thus, the result of Theorem 4 holds even when one requires the learners to be consistent, for any non-decreasing unbounded recursive h which is ≥ 1 on all inputs.*

However, the consistent learner, as in the above remark, outputting N from time to time, may not be class preserving. In case one requires class preserving consistency, the following theorem holds.

Theorem 8 *Suppose $L_i = \{\langle x, b_x \rangle : x \in N\}$, where b_r is the $(r+1)$-th least significant bit of i in binary representation (the least significant bit is b_0).*

Let $\mathcal{L} = \{L_i : i \in N\}$.

Then,

(a) $\mathcal{L}$ can be class-preservingly consistently learnt by a learner $\mathbf{M}$ which makes at most $\lceil \log(i+1) \rceil$ mind changes on L_i;

(b) For any class-preserving consistent learner $\mathbf{M}$ for $\mathcal{L}$, $\mathbf{F_M}(n) \geq \lceil \log(n+1) \rceil$.

Proof. (a) Consider a learner which, on input σ, outputs a grammar for L_i, where i is the least number such that L_i is consistent with σ. Then, it is easy to verify that the above learner satisfies (a).

(b) Consider any learner $\mathbf{M}$ for $\mathcal{L}$ which is consistent. Suppose any m is given. Then, let b_x, for $x < m$, be defined as follows: if $W_{\mathbf{M}(\langle 0,b_0\rangle,\langle 1,b_1\rangle,\ldots,\langle x-1,b_{x-1}\rangle)}$ contains $\langle x, 0\rangle$, then $b_x = 1$; otherwise $b_x = 0$. Then we have that for some i, which can be written using at most m bits, $\mathbf{M}$ makes at least m mind changes for consistently learning L_i. Thus $\mathbf{F_M}(n)$ is at least $\lceil \log(n+1)\rceil$. ▮

Now we consider what happens if the conditions of Theorem 4 do not hold. First, we consider the case when decidability of the equivalence problem still holds, but subset chains of length more than 1 are allowed.

Proof of the following Theorem 9 essentially exploits the following idea. Note that for any infinite set B and finite sequence σ, if content$(\sigma) \subseteq B$, and a learner learns both B and B', a finite subset of B containing content(σ), then the learner makes a mind change, beyond σ, on some text for B extending σ. For each learner $\mathbf{M}_i$ the proof uses a set L_{2i} (representing B above). It then constructs $\sigma_0, \sigma_1, \ldots, \sigma_r$, with $r \leq h(2i)$, by potentially placing a finite subset of L_{2i} containing content(σ_j) into the class $\mathcal{L}$ in order to force $h(2i)$ mind changes by $\mathbf{M}_i$ (in case $\mathbf{M}_i$ learns $\mathcal{L}$). It will be the case that at most one of the above finite sets is actually placed in $\mathcal{L}$ and others are spoiled (by making them non-subset of L_{2i}), thus satisfying the requirement of having a subset chain of length at most 2.

Theorem 9 *Suppose h is any recursive increasing function. There exists an indexed family $\mathcal{L}$, where the indexing is one-to-one, for which there is no subset chain of length more than 2, and there is no speedup. That is,*

(a) $\mathcal{L}$ can be **TxtEx***-learnt, using a class preserving hypothesis space, by a learner $\mathbf{M}$, such that $\mathbf{F_M}(i) \leq h(i) + 1$.*

(b) For any $\mathbf{M}$ which **TxtEx***-identifies $\mathcal{L}$, $\mathbf{F_M}(2i) \geq h(2i)$.*

(Actually, the condition of not having a subset chain of length more than 2 can be made even stronger: If $L \subset L'$ are in the class, then L and L' are incomparable to every other language in the class).

Proof. Let $L_{2i} = \{\langle i, x\rangle : x \text{ is odd}\}$.

Let $A_i^r = \{\langle i, x\rangle : x \text{ is odd and } x \leq r\}$.

Let $B_i^{r,y} = \{\langle i, x\rangle : x \text{ is odd and } x \leq r\} \cup \{\langle i, 2\langle r, y\rangle\rangle\}$.

We let $\mathcal{L} = \{L_j : j \in N\}$, where L_j, for odd j, are defined below. They will be of the form A_i^r or $B_i^{r,y}$ which are chosen to be in the class based on the following construction (where it is ensured that if $L_j = A_i^r$ or $B_i^{r,y}$, then $j \geq 2i$ and $r \geq 1$).

It will be the case that, for any i, r, $\mathcal{L}$ contains at most one of A_i^r or $B_i^{r,y}$ (for some y), and there are at most $h(2i)+1$ many different r's such that $\mathcal{L}$ contains A_i^r or $B_i^{r,y}$ (for some y).

We now give the construction for L_j, for j being odd. The following process is run in dovetailing fashion for each i. For a given i, the languages constructed below are of the form A_i^r or $B_i^{r,y}$. These are used to diagonalize against the learner $\mathbf{M}_i$.

All the processes have a common atomic procedure $New()$. This procedure, when called with parameter i, returns the least odd number $j \geq 2i$ such that j has not been returned by any other New() call before (to any process). Note that for every i there is at least one call $New(i)$ by the following process; thus for each odd j, L_j gets defined.

```
Given $i$, construction for the languages in $\mathcal{L}$ which are of the form $A_i^r$ or $B_i^{r,y}$,
  for some $r, y$.
  Initially, let $\sigma_0 = \langle i, 1\rangle$.
  For $k = 0$ to $h(2i)$ do:
  1.   Let $w = \max(\{w' : \langle i, w'\rangle \in \text{content}(\sigma_k)\})$.
  2.   Let $j = New(i)$.
       We now define the language $L_j$ which gets added to $\mathcal{L}$. Initially, $L_j$
         is $A_i^w$. Define more and more elements not in $A_i^w$ to be not in $L_j$
         until step 3 succeeds. If and when step 3 succeeds, go to step 4.
  3.   Search for a $\tau$ extending $\sigma_k$ such that content$(\tau) \subseteq L_{2i}$, and $\mathbf{M}_i(\sigma_k) \neq$
         $\mathbf{M}_i(\tau)$.
  4.   Let $L_j = B_i^{w,y}$, for an even $y$ such that $L_j(\langle i, 2\langle w, y\rangle\rangle)$ has not been
         defined upto now and $y > w$.
  5.   Let $\sigma_{k+1}$ be an extension of $\tau$ such that content$(\sigma_{k+1}) = A_i^{w'}$, for
         some odd $w' > w$ such that $w'$ bounds the time/steps needed by the
         underlying dovetailing process to get upto here in the construction.
  EndFor
```

Note that if $\mathbf{M}_i$ **TxtEx**-learns $\mathcal{L}$, then the search in step 3 will succeed. Furthermore, only the last incomplete iteration of the "for" loop may generate a subset of L_{2i}. All other languages generated are incomparable to each other. Thus the languages in $\mathcal{L}$ satisfy the "subset" constraints of the theorem.

Furthermore, note that, if $\mathbf{M}_i$ **TxtEx**-learns $\mathcal{L}$, then the above construction forces at least $h(2i)$ many mind changes for $\mathbf{M}_i$ on some text for L_{2i} (one for each value of k in $\{1, 2, \ldots, h(2i)\}$; note that the initial conjecture when $k = 0$, may not be a mind change, as the $\mathbf{M}_i$ may start with ?). Thus, the condition (b) of the theorem holds.

To see (a), let $g(i,k)$ denote a program which decides L_j, for the j as in iteration k, step 2, of the for loop above if iteration k exists; otherwise it is a program which decides L_{2i}. Note that such a grammar can be easily defined, as one can slowly follow L_{2i}, and if one observes iteration k to have started, then follow L_j as in there — see step 5 in the construction above which allows us to do this.

Now, if $\langle i, w\rangle$ is the largest element seen in the input so far and w is even, then the learner immediately knows the input language and can output a grammar appropriately. On the other hand, if w is odd, then the learner simulates the construction above for w time/steps to find the largest k such that the construction above (in the process with parameter i), after w steps, reaches iteration k in the loop (where, we take k to be 0 if the construction above has not reached the start of the For loop). Now, if $\langle i, w\rangle$ belongs to L_j, where j is as in iteration k of the loop in the construction above, then the learner outputs $g(i,k)$. Otherwise, it outputs $g(i,k+1)$.

It is easy to verify that the learner above **TxtEx**-learns the class $\mathcal{L}$ and makes at most $h(k)+1$ mind changes on a text for the language L_k. ∎

Now we will study what can happen if the languages in an indexed class are equal or incomparable, but the equivalence problem may be undecidable. We begin with the following useful technical proposition.

Proposition 10 *There exists a recursive function G such that, given any m, n, e, ℓ, σ such that $\ell \leq |\sigma|$, and $\mathrm{content}(\sigma) \subseteq \{\langle e,x\rangle : \langle e,x\rangle \leq m\}$, $G(e,m,n,\ell,\sigma)$ is a set S of 2^n indices for decision procedures such that*

(i) for each $j \in S$, $\mathrm{content}(\sigma) \subseteq \varphi_j^{-1}(1) \subseteq \mathrm{content}(\sigma) \cup \{\langle e,x\rangle : \langle e,x\rangle > m\}$;

(ii) for all $j, j' \in S$, $\varphi_j^{-1}(1)$ and $\varphi_{j'}^{-1}(1)$ are either equal or incomparable;

*(iii) if $\mathbf{M}_e$ **TxtEx**-identifies $\varphi_j^{-1}(1)$ for each $j \in G(e,m,n,\ell,\sigma)$, and $W_{\mathbf{M}_e(\sigma[\ell])} \cap \{y : y \leq m\} \neq \mathrm{content}(\sigma)$, then for some $j \in S$, for some text T for $\varphi_j^{-1}(1)$, which starts with σ, $\mathrm{card}(\{k \geq \ell : \mathbf{M}_e(T[k]) \neq \mathbf{M}_e(T[k+1])\}) \geq n+1$.*

Proof. Let G be defined as follows. For any suitable parameters, suppose we have defined $G(\cdot,\cdot,n',\cdot,\cdot)$, for $n' < n$. Then, we define $G(e,m,n,\ell,\sigma)$ as follows (where the parameters satisfy the hypothesis of the proposition). By implicit use of effective s-m-n theorem, $G(e,m,n,\ell,\sigma)$ is a set S of 2^n indices which behave as follows. Assume, $\ell \leq |\sigma|$, and $\mathrm{content}(\sigma) \subseteq \{\langle e,x\rangle : \langle e,x\rangle \leq m\}$ (otherwise, $G(e,m,n,\ell,\sigma)$ can be defined so that $\mathrm{content}(\sigma) = \varphi^{-1}(j)$ for each $j \in G(e,m,n,\ell,\sigma)$).

If $n = 0$, then we let $G(e,m,n,\ell,\sigma)$ enumerate a set S of one index which is a decision procedure for content(σ).

Otherwise, (i.e., $n > 0$), initially, for each $j \in S$, for $y \leq \langle e, m+1\rangle$, let $\varphi_j(y) = 1$ iff $y \in \text{content}(\sigma) \cup \{\langle e, m+1\rangle\}$. Then, for each $j \in S$, for larger and larger $y > \langle e, m+1\rangle$, we let $\varphi_j(y) = 0$, until a σ' extending σ is found such that content(σ') = content$(\sigma) \cup \{\langle e, m+1\rangle\}$, and $\mathbf{M}_e(\sigma') \neq \mathbf{M}_e(\sigma[\ell])$. If and when such a σ' is found let $m' = \max(\{y : \varphi_j(y)$ has been defined so far for some $j \in S\})$. Let σ_1 be an extension of σ' such that content(σ_1) = content$(\sigma) \cup \{\langle e, m+1\rangle, \langle e, m'+1\rangle\}$. Let σ_2 be an extension of σ' such that content(σ_2) = content$(\sigma) \cup \{\langle e, m+1\rangle, \langle e, m'+2\rangle\}$. Let 2^{n-1} members of S follow the 2^{n-1} members of $G(e, \langle e, m'+2\rangle, n-1, |\sigma'|, \sigma_1)$ and the remaining 2^{n-1} members of S follow the 2^{n-1} members of $G(e, \langle e, m'+2\rangle, n-1, |\sigma'|, \sigma_2)$. Note that $W_{M_e(\sigma')} \cap \{y : y \leq \langle e, m'+2\rangle\}$ is not equal to at least one of content(σ_1) and content(σ_2). It is now easy to verify by induction that G satisfies the requirements of the proposition. ∎

Using the above proposition, we can now show that, for some class with undecidable equivalence problem and equal or incomparable languages, the lower bound on $\mathbf{F}_{\mathbf{M}}(n)$ is at least $\log n - o(\log n)$, and, thus, AST cannot hold for such classes.

Proofs of the next two theorems are based on techniques used in [BF74,FBP91] for similar theorems for function learning.

Theorem 11 *Given any non-decreasing recursive function f with unbounded range, there exists an indexed family $\mathcal{L} = (L_i)_{i\in N}$, where, for all j and k, either $L_j = L_k$ or L_j and L_k are incomparable, such that for all* **M TxtEx**-*identifying $\mathcal{L}$, $\mathbf{F}_{\mathbf{M}}(n) \geq \log n - f(n)$ for infinitely many n.*

Proof. Without loss of generality assume that $\mathbf{M}(\Lambda) =?$ for the given learner $\mathbf{M}$. Let $n_0 = 0$, and $n_{i+1} = n_i + 2^r$ such that $r > \log n_{i+1} - f(n_{i+1})$. Let G be as in Proposition 10. For each i, let L_j, $n_i \leq j < n_{i+1}$, follow the $n_{i+1} - n_i$ languages $\varphi_k^{-1}(1)$, $k \in G(\pi_1(i), 0, \log(n_{i+1} - n_i), 0, \Lambda)$ (which need not be distinct). Thus by Proposition 10, for all i, $\mathbf{F}_{\mathbf{M}_{\pi_1(i)}}(n_{i+1}) \geq \log n_{i+1} - f(n_{i+1})$. Now, as $\mathbf{M}$ is equivalent to some $\mathbf{M}_p$, we have that for infinitely many i, such that $\pi_1(i) = p$, $\mathbf{F}_{\mathbf{M}}(n_{i+1}) = \mathbf{F}_{\mathbf{M}_{\pi_1(i)}}(n_{i+1}) \geq \log n_{i+1} - f(n_{i+1})$. Theorem follows. ∎

The next theorem shows that, yet, every indexed class with equal or incomparable languages can be learned using approximately $\log n$ mind changes.

Theorem 12 *Every indexed family $\mathcal{L} = (L_i)_{i\in N}$, such that for all i, j, either $L_i = L_j$ or L_i and L_j are incomparable, can be* **TxtEx**-*learnt by a learner* **M**, *using a class preserving hypothesis space, such that $\mathbf{F}_{\mathbf{M}}(n) \leq \log n + \log\log n + o(\log\log n)$.*

(Here, for ease of notation, we take $\log n$ and $\log\log n$ to be 1, for $n \leq 2$). PROOF. Let

$$p_n = \frac{c}{n \log n (\log\log n)^2}$$

where $c > 0$ is such that $\sum_n p_n \leq 1$, and we take $\log n$ and $\log\log n$ to be 1 for $n \leq 2$.

Let w_n be a rational number, effectively computed from n such that $p_n/2 \leq w_n \leq p_n$.

Let $wt(S) = \sum\{w_j : j \in S\}$. Let

$$\alpha_i = \frac{1}{2^i} + \sum_{j=0}^{i} \frac{1}{2^{i+2j}}$$

$$r_i = \frac{1}{2^{3i+2}}$$

Note that $\alpha_{i+1} = \frac{1}{2}(\alpha_i + r_i)$.

We think of a learner $\mathbf{M}$ as having been given the whole text T, and working in stages, outputting conjectures in each stage.

Initially, $S_0 = \emptyset$, and $i = 0$.

Inductively, the following invariants will be satisfied.

(IA) for all $j \in S_i$, content$(T) \not\subseteq L_j$.

(IB) $wt(N - S_i) \leq \alpha_i$.

For any finite set S, let $majgram(S)$ denote the majority weight follower among decision procedures in S. That is, (a) $majgram(S)$ is a decision procedure for some L_j, $j \in S$, and (b) if there exists a $S' \subseteq S$, such that $wt(S') > wt(S)/2$, and for all $j, j' \in S'$, $L_j = L'_j$, then $majgram(S)$ is a decision procedure for L_j, for some $j \in S'$.

At stage i (starting with stage 0), $\mathbf{M}$ first searches for a $X_i \subseteq N - S_i$, such that $wt(N - S_i - X_i) \leq r_i$. Then, the learner outputs $majgram(X_i)$. It then waits until it finds a subset $Y_i \subseteq X_i$ such that $wt(Y_i) \geq \frac{1}{2}(wt(X_i))$, and each $j \in Y_i$ satisfies content$(T) \not\subseteq L_j$. At which point the learner computes $S_{i+1} = S_i \cup Y_i$, and goes to stage $i + 1$.

It is easy to verify that (IA) and (IB) are satisfied.

Now consider any L_k and text T for L_k. Clearly, the learner converges, as $wt(\{k\}) \geq \frac{c}{2k \log k (\log\log k)^2}$, and thus, by (IA) and (IB), we cannot have infinitely many stages. Furthermore, for the last stage i that is executed, we must have that there is no $Y_i \subseteq X_i$ such that $wt(Y_i) \geq \frac{1}{2}wt(X_i)$ and for all $j \in Y_i$, content$(T) \not\subseteq L_j$. It immediately follows that $wt(Z_i) > \frac{1}{2}(wt(X_i))$, where $Z_i = \{j \in X_i : L_k = L_j\}$. Thus, $majgram(X_i)$ is a decision procedure for L_k.

Thus, **M** **TxtEx**-identifies $\mathcal{L}$.

Furthermore, the number of stages i executed by the learner (and, thus, the number of mind changes made by **M** on T) satisfies $\frac{7}{3*2^i} \geq \alpha_i \geq wt(\{k\}) \geq w_k \geq \frac{c}{2k \log k (\log\log k)^2}$. Thus, $i \leq \log k + \log\log k + o(\log\log k)$. ▮

Furthermore, one can modify the construction in the above theorem to bound $\mathbf{F_M}(n)$ by $\log n + \log\log n + \ldots + o(\log\log\log \ldots \log n)$, by using

$$p_n = \frac{c}{n \log n (\log\log n) \ldots (\log\log\log \ldots \log n)^2}.$$

Note that the above result does not hold if one requires, on the positive side, that the learner uses the given indexing of $\mathcal{L}$ as the hypothesis space. This follows from the corresponding result for function learning from [Bār74b].

4 Automatic Classes and Learning

In this section, we introduce necessary concepts for automatic learning of automatic classes.

Let Σ denote a non-empty finite alphabet. Let Σ^* denote the set of all strings over the alphabet Σ. Let ϵ denote the empty string. We let $|w|$ denote the length of string w. We fix some arbitrary order among the members of Σ. For strings x and y, $x <_{lex} y$ denotes that x is lexicographically (that is, in dictionary order) before y. The relation $x <_{ll} y$ denotes that x is length-lexicographically before y, that is, either $|x| < |y|$, or $|x| = |y|$ and $x <_{lex} y$. When we consider sets of strings, $\min(S)$ and $\max(S)$ denote the length-lexicographically minimal and maximal strings in S, where $\max(\emptyset) = \epsilon$ and $\min(\emptyset)$ is undefined. We let $\mathrm{succ}_L(w)$ and $\mathrm{pred}_L(w)$ denote the successor and predecessor of w in the length-lexicographical ordering of the language L, where $\mathrm{pred}_L(w)$ is undefined for the length-lexicographically least string in L, and $\mathrm{succ}_L(w)$ is undefined for the length-lexicographically maximal string in L (if any). For a given Σ and $w \in \Sigma^*$, let $ord(w)$ denote the number of strings in Σ^* which are $<_{ll} w$. We let cf_L denote the characteristic function of L.

The *convolution* (see [KN95]) of two strings $x, y \in \Sigma^*$, $\mathrm{conv}(x, y)$, is defined as the string $(x(0), y(0))(x(1), y(1)) \ldots (x(n-1), y(n-1))$, where each pair is a symbol from $(\Sigma \cup \{\diamond\})^2$ and $n = \max(|x|, |y|)$. The special symbol $\diamond \notin \Sigma$

is appended (as many times as needed) to the shorter string in order to make both strings to be of the same length n. Similarly, conv can be defined on multiple arguments. An n-ary relation R or an m-ary function f is called *automatic* if the sets $\{\mathrm{conv}(x_1, x_2, \ldots, x_n) : R(x_1, x_2, \ldots, x_n)\}$ and $\{\mathrm{conv}(x_1, x_2, \ldots, x_m, y) : f(x_1, x_2, \ldots, x_m) = y\}$, respectively, are regular.

A family of languages over alphabet Σ, $\{L_\alpha : \alpha \in I\}$ is said to be *automatic* (see [KN95]) iff I is a regular set, $L_\alpha \subseteq \Sigma^*$ for each $\alpha \in I$, and $\{\mathrm{conv}(\alpha, x) : x \in L_\alpha\}$ is regular.

When we are considering learning of automatic classes, the elements of languages are strings rather than natural numbers. Most of the definitions and notations discussed above for learning languages over natural numbers carry over to the case of learning languages over strings, with numbers being replaced by strings; we omit the details. Below we describe a special kind of learner, called *automatic* learner ([JLS10,CJO$^+$11]). An automatic learner is an automatic mapping from previous memory and current datum to new memory and new conjecture. That is, the relation (previous memory, current datum, new memory, new conjecture) is automatic. Here memory is a string over some alphabet Γ. Suppose T is the input text for the automatic learner $\mathbf{Q}$. Let $(mem^T_{n+1}, hyp^T_{n+1}) = \mathbf{Q}(mem^T_n, T(n))$, where mem^T_0 and hyp^T_0 are some default initial memory mem_0 and the default initial hypothesis hyp_0 of the learner $\mathbf{Q}$. We can consider the hypothesis hyp^T_n of the learner $\mathbf{Q}$ as its output on the input $T[n]$, and thus the learnability notions discussed in Section 2.1 above can be taken over to the setting of automatic learners. Below we let $\mathbf{Q}$ range over automatic learners. Here are some examples:

(a) Let $\Sigma = \{0, 1\}$. Let $L_\alpha = \{x : \alpha \leq_{ll} x\}$. Then, $\mathcal{L} = \{L_\alpha : \alpha \in \{0, 1\}^*\}$ is automatically learnable. The learner just remembers the length-lexicographically least string in the input text (which is also its conjecture).

(b) Let $\Sigma = \{0, 1\}$. Let $L_{\mathrm{conv}(\alpha,\beta)} = \{x : \alpha \leq_{ll} x \leq_{ll} \beta\}$, for $\alpha, \beta \in \{0, 1\}^*$. Then, $\mathcal{L} = \{L_{\mathrm{conv}(\alpha,\beta)} : \alpha, \beta \in \{0, 1\}^*\}$ is automatically learnable. The learner just remembers (a convolution of) the length-lexicographically least and length-lexicographically largest element in the input text (this convolution is also its conjecture).

(c) Let $\Sigma = \{0, 1\}$. Let $L_\alpha = \{x : |x| = |\alpha|, x \neq \alpha\}$. Then, $\mathcal{L} = \{L_\alpha : \alpha \in \{0, 1\}^*\}$ is not automatically learnable [JLS10].

When dealing with automatic families, we let $\mathbf{F_M}(w)$ = maximum over the mind changes made by the learner $\mathbf{M}$ on any input text for a language L_u, $u \leq_{ll} w$.

Note that for learning automatic families, as long as memory is not restricted (except due to the definition of automatic learner), one can assume the hypothesis space to be the same as the automatic class being learnt (this holds as one can decide whether $H_i = H'_j$, for any given automatic families $(H_i)_{i \in I}$ and $(H'_j)_{j \in I'}$). Thus, for the next section, for all the results the hypothesis space used is the automatic family being learnt. Furthermore, one can also assume, without loss of generality, that the automatic class is one-to-one (that is, it has at most one index for any language).

5 Mind Change Speed-up for Automatic Classes

In the sequel, pairing is assumed to be done via convolution, that is, for strings $x_1, x_2, \ldots$, $(x_1, x_2, \ldots)$ is taken as $\text{conv}(x_1, x_2, \ldots)$. We begin with an example of an automatic class containing languages over the unary alphabet with linear lower and upper bounds on the number of mind changes.

Theorem 13 *Let $L_{0^i} = \{0^j : j < i\}$. Let $\mathcal{L} = \{L_{0^i} : i \in N\}$. Then,*

(a) $\mathcal{L}$ can be **TxtEx***-learnt by an automatic learner* **Q** *such that $\mathbf{F_Q}(0^n) = n$.*

(b) Any learner **M** *which* **TxtEx***-learns $\mathcal{L}$ has $\mathbf{F_M}(0^n) \geq n$.*

Proof. (a) Consider a learner **Q** which starts with intial conjecture and initial memory 0^0. It will be the case that the memory and conjecture of **Q** will always be the same. **Q**, on previous memory 0^j and current datum 0^i, has new memory and new conjecture as $0^{\max(\{i+1,j\})}$. It is easy to verify that **Q** witnesses (a).

(b) For any **TxtEx**-learner **M** for $\mathcal{L}$, one can construct σ_i, $i \in N$, such that $\sigma_i \subseteq \sigma_{i+1}$, $\text{content}(\sigma_i) = L_{0^i}$, and $\mathbf{M}(\sigma_i)$ is a grammar for L_{0^i}. Then $\text{MC}_\mathbf{M}(\sigma_i) \geq i$. ∎

Now we show that, for any automatic function h (with the range containing strings over a unary alphabet), there is an automatic class that can be learned automatically with h (more precisely, $ord(h(0^{i+1}, \epsilon) + 1)$ being the tight bound on the number of mind changes).

Theorem 14 *Suppose h is a non-decreasing automatic function with* $\text{range}(h) \subseteq 0^+$. *Let*

$L_{(0^{i+1}, \epsilon)} = \{(0^{i+1}, 1^j) : j \in N\}$,

$L_{(0^{i+1}, 1^{j+1})} = \{(0^{i+1}, 1^r) : r < j + 1\}$,

$L_{(\epsilon, \epsilon)} = \emptyset$, *and*

$\mathcal{L} = \{L_{(\epsilon,\epsilon)}\} \cup \{L_{(0^{i+1},1^j)} : i \in N, j \leq ord(h(0^{i+1},\epsilon))\}$.

Then,

(a) $\mathcal{L}$ can be **TxtEx**-*learnt by an automatic learner* $\mathbf{Q}$, *such that* $\mathbf{F_Q}(0^{i+1},\epsilon) \leq ord(h(0^{i+1},\epsilon)) + 1$.

(b) Any learner $\mathbf{M}$ *which* **TxtEx**-*learns* $\mathcal{L}$ *has* $\mathbf{F_M}(0^{i+1},\epsilon) \geq ord(h(0^{i+1},\epsilon)) + 1$.

Proof. (a) Let $\mathbf{Q}$ be a learner which starts with initial memory and initial conjecture as (ϵ,ϵ). It will be the case that the memory and conjecture of $\mathbf{Q}$ will always be the same. The learner $\mathbf{Q}$ does not change its memory/conjecture on any input datum which is not of the form $(0^{i+1},1^j)$. The new memory and conjecture of $\mathbf{Q}$ on current datum $(0^{i+1},1^j)$ is defined as follows:

- if $j \geq ord(h(0^{i+1},\epsilon))$ or previous memory is $(0^{i+1},\epsilon)$, then new memory and conjecture are $(0^{i+1},\epsilon)$;
- if $j < ord(h(0^{i+1},\epsilon))$, and previous memory is $(0^{i+1},1^{j'+1})$, with $j' \leq j$, or previous memory is (ϵ,ϵ), then new memory and conjecture are $(0^{i+1},1^{j+1})$;
- new memory and conjecture are same as the previous memory, otherwise. (This case is taken even for the case when the current datum is #.)

It is easy to verify that $\mathbf{Q}$ satisfies the requirements. Here, note that the number of mind changes made by $\mathbf{Q}$ on input $L_{(0^{i+1},1^j)}$, for $1 \leq j \leq ord(h(0^{i+1},\epsilon))$, is bounded by j.

(b) Follows as learning $L_{0^{i+1},\epsilon}$ will need $ord(h(0^{i+1},\epsilon)) + 1$ mind changes due to $\{\emptyset\} \cup \{L_{(0^{i+1},1^j)} : i \in N, 1 \leq j \leq ord(h(0^{i+1},\epsilon))\} \subseteq \mathcal{L}$. ∎

Now our goal is to show that, under certain natural conditions, mind change speed-up for automatic classes is possible if an automatic learner uses fat texts.

Proofs of Theorems 15 and 16 are the most difficult in this paper. In these theorems, on one hand, the class $\mathcal{L}$ considered is automatic (so equivalence, subset problem, etc., among languages in the class are decidable), but, on the other hand, the learner is automatic and we also allow some subset relations among languages. The main difficulty is because of the learner being automatic, thus forgetting past data. The proof again uses delaying of mind change until it is safe, by cancelling all but c wrong grammars upto some large enough bound (in a way similar to Proof for Theorem 4). Here c is a constant such that at most c different languages in the class are related by subset/superset relation with any particular language of the class. Then, the learner finds upto c^2 many grammars which may be for the input language, in case any of the languages, with indices below the large enough bound mentioned above, contains the input language. The learner then proceeds to try these languages one by one (where smaller languages are tried first). Due to forgetting of past data by automatic learners, one needs a fat text to be able to cancel out wrong grammars. Formal details follow. The proof for arbitrary speed-up (Theorem 16) is technically involved, and thus we begin by showing a simpler version first.

Theorem 15 *Suppose $\mathcal{L} = \{L_\alpha : \alpha \in I\}$ is an automatic family (without loss of generality, assume one-to-one). Suppose constants k and c are given, where for all $L \in \mathcal{L}$, $\mathrm{card}(\{L' \in \mathcal{L} : L \subseteq L'$ or $L' \subseteq L\}) \leq c$.*

Then, there exists an automatic learner $\mathbf{Q}$ which learns $\mathcal{L}$ from fat texts such that (for learning from fat texts) $\mathbf{F_Q}(\alpha) \leq \max(\{\lceil |\alpha|/k \rceil, 1\}) * c^2 - 1$.

Proof. Without loss of generality assume that there are at least $c+1$ indices of length at most k. The learner Q defined below operates in phases. Intuitively, memory of $\mathbf{Q}$ is of the form

$$(0^i, 0^p, \alpha_1, \alpha_2, \ldots, \alpha_{c+1}, \beta_1, \beta_2, \ldots, \beta_{c^2}, \mathit{prevconj}),$$

where

- (i) $p = k * i$;
- (ii) $\alpha_j <_{ll} \alpha_{j+1}$, for $1 \leq j < c$;
- (iii) $\alpha_c \leq_{ll} \alpha_{c+1}$;
- (iv) $\mathit{prevconj}$ is the previous conjecture;
- (v) $\mathbf{Q}$ has already made $(i-1)$-phases (each producing upto c^2 conjectures), and is now in its i-th phase;
- (vi) for all α such that $|\alpha| \leq p$ and $\alpha \notin \{\alpha_j : 1 \leq j \leq c\} \cup \{\gamma : \alpha_c \leq_{ll} \gamma \leq_{ll} \alpha_{c+1}\}$, $\mathbf{Q}$ has already observed a string in the input which is not in L_α;
- (vii) in case $\alpha_c = \alpha_{c+1}$, $\beta_1, \ldots, \beta_{c^2}$ denote the c^2 possible members β of I such that L_β is contained in one of L_{α_j}, $1 \leq j \leq c$ (in case of $< c^2$ such members, we use # for the remaining elements); furthermore, if $L_{\beta_j} \subseteq L_{\beta_{j'}}$, then $j \leq j'$;
- (viii) in case $\alpha_c = \alpha_{c+1}$, $\mathit{prevconj} = \beta_j$ for some j such that $1 \leq j \leq c^2$, and for $1 \leq j' < j$, $\mathbf{Q}$ has already observed a string in the input which is not in $L_{\beta_{j'}}$.

Initially, the memory of $\mathbf{Q}$ is $(0^1, 0^k, \alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_c, \alpha_{c+1}, \#, \#, \ldots, \#, ?)$, where α_{c+1} is the length-lexicographically largest element of I of length at most k, and $\alpha_1, \ldots, \alpha_c$ are the c length-lexicographically least elements of I. The initial conjecture of $\mathbf{Q}$ is ?.

At any point during the learning process, if the new input datum is w and the previous memory is $(0^i, 0^p, \alpha_1, \alpha_2, \ldots, \alpha_{c+1}, \beta_1, \beta_2, \ldots, \beta_{c^2}, \mathit{prevconj})$, then $\mathbf{Q}$ behaves as follows:

- (1.) If $\alpha_c \neq \alpha_{c+1}$, and $w \notin L_{\alpha_j}$ for some least j with $1 \leq j \leq c+1$, then
  - (1.1.) If $j = c+1$, then let $\alpha'_{c+1} = \text{pred}_I(\alpha_{c+1})$, and $\alpha'_r = \alpha_r$ for $1 \leq r \leq c$; otherwise, let (i) $\alpha'_r = \alpha_r$, for $1 \leq r < j$, (ii) $\alpha'_r = \alpha_{r+1}$, for $j \leq r < c$, (iii) $\alpha'_c = \text{succ}_I(\alpha_c)$, and (iv) $\alpha'_{c+1} = \alpha_{c+1}$.
  - (1.2.) If $\alpha'_c \neq \alpha'_{c+1}$, then let $\beta_1 = \ldots = \beta_{c^2} = \#$, and let new memory be $(0^i, 0^p, \alpha'_1, \ldots, \alpha'_{c+1}, \beta_1, \ldots, \beta_{c^2}, \mathit{prevconj})$ and let new conjecture be $\mathit{prevconj}$.
  - (1.3.) else (i.e., $\alpha'_c = \alpha'_{c+1}$), let $\beta_1, \ldots, \beta_{c^2}$ denote the c^2 possible members β of I such that L_β is contained in one of $L_{\alpha'_j}$, $1 \leq j \leq c$; furthermore, if $L_{\beta_j} \subseteq L_{\beta_{j'}}$, then $j \leq j'$; If there are several possible orders to choose β_j satisfying the above, then choose the lexicographically least order among them. (In case of $< c^2$ members β of I such that L_β is contained in some L_{α_j}, we use # for the remaining β's); Conjecture β_1, and let new memory be $(0^i, 0^p, \alpha'_1, \ldots, \alpha'_{c+1}, \beta_1, \ldots, \beta_{c^2}, \beta_1)$.
- (2.) else (if $\alpha_c = \alpha_{c+1}$), then
  - if $w \notin L_{\mathit{prevconj}}$, then

    (2.1.) if $\mathit{prevconj} = \beta_j$, and $j < c^2$ and $\beta_{j+1} \neq \#$, then let new memory be $(0^i, 0^p, \alpha_1, \alpha_2, \ldots, \alpha_{c+1}, \beta_1, \beta_2, \ldots, \beta_{c^2}, \beta_{j+1})$ and the new conjecture be β_{j+1}.

    (2.2.) otherwise, let new memory be

    $$(0^{i+1}, 0^{p+k}, \alpha'_1, \alpha'_2, \ldots, \alpha'_{c+1}, \#, \#, \ldots, \#, \mathit{prevconj}),$$

    where α'_{c+1} is the length-lexicographically largest element of I of length at most $p+k$, and $\alpha'_1, \ldots, \alpha'_c$ are the c length-lexicographically least elements of I. Conjecture $\mathit{prevconj}$.
  - else (i.e., $w \in L_{\mathit{prevconj}}$) repeat the old memory and conjecture.
- (3.) else (i.e., $\alpha_c \neq \alpha_{c+1}$, and $w \in L_{\alpha_j}$ for all j with $1 \leq j \leq c+1$) repeat the old memory and conjecture.

Intuitively, for any w, in step (1) the learner (over several inputs) tries to eliminate all but c of the potential conjectures of length at most p; all the eliminated conjectures do not contain the input language (see steps 1, 1.1 and 1.2). Once the learner is left with only c conjectures of length at most p, which may contain the input language, it finds the indices of all the potential c^2 many languages which may be for the input language (unless none of the languages, with index of length at most p, contain the input language) (see step 1.3).

After this, in steps 1.3, 2 and 2.1, the learner serially tries all the above c^2 many languages which could be the input language. (Note that, the testing of these languages is done in a specific order so that subsets are tried earlier than the supersets.) Then, the learner eliminates them one by one, until it finds the correct language or observes that none of them could contain the input language (i.e., all languages in $\mathcal{L}$ which contain the input have indices of length larger than p). In which case the learner goes to the next ($i+1$-th) phase (step 2.2).

It is now easy to verify that the above learner **TxtEx**-identifies $\mathcal{L}$ on fat texts, and on L_α makes at most $\max(\{\lceil|\alpha|/k\rceil, 1\}) * c^2 - 1$ mind changes (using $\max(\{\lceil|\alpha|/k\rceil, 1\})$ phases, each of which may make upto c^2 conjectures). ∎

Note that the above proof uses fat texts to be able to check whether a language in the automatic family contains the input language or not.

In the above theorem, one can replace c^2 by c, if, instead of using conjectures β_j one by one, the learner (i) keeps track of β_j such that it hasn't seen a non-element of L_{β_j}, and (ii) outputs a conjecture β_j if the learner hasn't seen a non-element of L_{β_j} and L_{β_j} is contained in every other $L_{\beta_{j'}}$ for which it hasn't seen a non-element. This ensures that in steps 1.3 and 2, for each value of i, at most c conjectures are output.

Furthermore, we can generalize the theorem above to beat (almost everywhere) mind changes given by any non-decreasing unbounded recursive function as follows. Suppose h is a recursive non-decreasing unbounded function. Let $H(i) = \min(\{j : h(j) > i\})$. Suppose M is a one tape Turing Machine which computes the mapping 0^i to $0^{H(i)}$. The snap shot of the computation done by the Turing Machine, at any point of time, can be given by its instantaneous description (ID) [HU79]. Note that, for a fixed TM M, the function to compute the next ID from the previous ID for TM M, $\mathit{nextID}_M(ID)$, is automatic. Now, instead of using memory as in the proof of the above theorem, we use memory of the form $(0^i, s, \mathit{temp}, \alpha_1, \alpha_2, \ldots, \alpha_{c+1}, \beta_1, \ldots, \beta_{c^2}, \mathit{prevconj})$, where initially, the memory is $(0^1, s_0, 1, \alpha_1, \alpha_2, \ldots, \alpha_{c+1}, \beta_1, \ldots, \beta_{c^2}, \mathit{prevconj})$, where s_0 is the initial ID of the TM M on input 0^1 (values of $\alpha_1, \alpha_2, \ldots,$ are irrelevant at the beginning). We have an additional step 0 in the construction. In case temp=1, this step computes $\mathit{nextID}_M(s)$; in the case that $\mathit{nextID}_M(s)$ is not a halting ID, the learner's new memory is updated to $(0^i, \mathit{nextID}_M(s), \mathit{temp}, \alpha_1, \alpha_2, \ldots, \alpha_{c+1}, \beta_1, \ldots, \beta_{c^2}, \mathit{prevconj})$; otherwise, first the learner determines 0^p, the content of the tape after M halts (this can be done using $\mathit{nextID}_M(s)$) and updates its memory to $(0^i, 0^p, 0, \alpha_1, \alpha_2, \ldots, \alpha_{c+1}, \beta_1, \ldots, \beta_{c^2}, \mathit{prevconj})$, where $\alpha_1, \ldots, \alpha_c$ are the length-lexicographically least members of I, α_{c+1} is the length-lexicographically largest element of I of the length at most p, and $\beta_1, \ldots, \beta_{c^2} = \#$. The remaining steps are the same as in the proof of the theorem above, except that they are applicable when $\mathit{temp} = 0$, and step 2.2. is updated to replace the new memory by, $(0^{i+1}, s, 1, \alpha_1, \alpha_2, \ldots, \alpha_{c+1}, \beta_1, \ldots, \beta_{c^2}, \mathit{prevconj})$, where s is the initial ID of M on input 0^{i+1} (value of $\alpha_1, \ldots, \beta_1, \ldots$ is irrelevant at this point, as they will be updated in step 0).

Thus, we have the following theorem.

Theorem 16 *Suppose $\mathcal{L} = \{L_\alpha : \alpha \in I\}$ is an automatic family (without loss of generality assume one-to-one). Suppose a non-decreasing unbounded recursive function h and a constant c are given, where for all $L \in \mathcal{L}$, $\mathrm{card}(\{L' \in \mathcal{L} : L \subseteq L'$ or $L' \subseteq L\}) \leq c$.*

Then, there exists an automatic learner $\mathbf{Q}$ which learns $\mathcal{L}$ from fat texts such that (for learning from fat texts) $\mathbf{F_Q}(\alpha) \leq \max(\{h(|\alpha|), 1\}) * c - 1$.

The above result also works if, instead of using fat texts, the languages in the class are required to be pairwise infinitely different (in addition to the requirement: for all $L \in \mathcal{L}$, $\mathrm{card}(\{L' \in \mathcal{L} : L \subseteq L'$ or $L' \subseteq L\}) \leq c$, for some constant c). For example, when the languages in the class are cylindrical, as in the case of each $L \in \mathcal{L}$ being of the form $\{(x, a) : x \in L', a \in 0^*\}$, for some corresponding L'. Thus, we have the following mind change speed-up result holding for automatic learning from arbitrary texts.

Theorem 17 *Suppose $\mathcal{L} = \{L_\alpha : \alpha \in I\}$ is an automatic family (without loss of generality assume one-to-one).*

Suppose non-decreasing unbounded recursive functions h and constant c are given, where for all $L \in \mathcal{L}$, $\mathrm{card}(\{L' \in \mathcal{L} : L \subseteq L'$ or $L' \subseteq L\}) \leq c$.

Furthermore, suppose that the languages in the class are pairwise infinitely different.

Then, there exists an automatic learner $\mathbf{Q}$ which learns $\mathcal{L}$ from texts such that $\mathbf{F_Q}(\alpha) \leq \max(\{h(|\alpha|), 1\}) * c - 1$.

Another case where the above result applies is when the alphabet used for the languages is of cardinality 1, that is $|\Sigma| = 1$ (in addition to the requirement that for all $L \in \mathcal{L}$, $\mathrm{card}(\{L' \in \mathcal{L} : L \subseteq L'$ or $L' \subseteq L\}) \leq c$, for some constant c). This holds, as for an alphabet of the size 1, an automatic learner can remember in the memory all strings seen [JLS10].

6 Conclusion

In 1972, Bārzdiņš and Freivalds introduced the maximum number of mind changes on the first n functions as a measure of efficiency of learning in the limit. Our interest in this measure of complexity for learning indexed classes of languages was revived by growing interest in automatic learning of automatic classes of languages. As mind change speed-up effects, discussed and resolved for learning recursive functions in [BKP74], surprisingly, have never been explored for learning languages from positive data, we, first, considered these issues for the corresponding framework. We also give a sufficient condition for a family of automatic classes for which speed-up is possible if either an automatic learner uses fat texts, or the languages in the classes in question differ infinitely. Yet the general problem of whether there are wide natural automatic classes for which mind change speed-up is possible remains open.

One can note that the mind change speed-up in both frameworks considered in our paper is achieved when a learner, choosing a new conjecture, accesses increasingly more data from the underlying numbering of languages. It would be very interesting to find out if the amount of such data can be measured in some form and what is the actual quantitative relationship between this amount and the number of mind changes.

Acknowledgements: A preliminary version of this paper appeared in STACS 2012 [JK12]. We thank the anonymous referees of this journal and STACS 2012 for several helpful comments.

References

[Ang80] D. Angluin. Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, 21(1):46–62, 1980.

[Bār74a] J. Bārzdiņš. Inductive inference of automata, functions and programs. In *Proceedings of the 20th International Congress of Mathematicians, Vancouver*, pages 455–460, 1974. In Russian. English translation in American Mathematical Society Translations: Series 2, 109:107-112, 1977.

[Bār74b] J. Bārzdiņš. Limiting synthesis of τ numbers. In *Theory of Algorithms and Programs, vol. 1*, pages 112–116. Latvian State University, Riga, Latvia, 1974. In Russian.

[Bei91] R. Beigel. Bounded queries to SAT and the boolean hierarchy. *Theoretical Computer Science*, 84:199–223, 1991.

[BF72] J. Bārzdiņš and R. Freivalds. On the prediction of general recursive functions. *Soviet Mathematics Doklady*, 13:1224–1228, 1972.

[BF74] J. Bārzdiņš and R. Freivalds. Prediction and limiting synthesis of recursively enumerable classes of functions. In *Theory of Algorithms and Programs, vol. 1*, pages 101–111. Latvian State University, Riga, Latvia, 1974. In Russian.

[BKP74] J. Bārzdiņš, E. Kinber, and K. Podnieks. Concerning synthesis and prediction of functions. In *Theory of Algorithms and Programs, vol. 1*, pages 117–128. Latvian State University, Riga, Latvia, 1974. In Russian.

[Blu67] M. Blum. A machine-independent theory of the complexity of recursive functions. *Journal of the ACM*, 14(2):322–336, 1967.

[CGH$^+$88] J Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V Sewelson, K Wagner, and G. Wechsung. The boolean hierarchy I: Structural properties. *SIAM Journal of Computing*, 17:1232–1252, 1988.

[CJO$^+$11] John Case, Sanjay Jain, Yuh Shin Ong, Pavel Semukhin, and Frank Stephan. Automatic learning with feedback queries. In *Models of Computation in Context, Seveth Conference on Computability in Europe (CiE)*, volume 6735 of *Lecture Notes in Computer Science*, pages 31–40. Springer-Verlag, 2011.

[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.

[CS83] J. Case and C. Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science*, 25:193–220, 1983.

[DS86] R. Daley and C. Smith. On the complexity of inductive inference. *Information and Control*, 69:12–40, 1986.

[FBP91] R. Freivalds, J. Bārzdiņš, and K. Podnieks. Inductive inference of recursive functions: Complexity bounds. In J. Bārzdiņš and D. Bjørner, editors, *Baltic Computer Science*, volume 502 of *Lecture Notes in Computer Science*, pages 111–155. Springer-Verlag, 1991.

[FKS95] R. Freivalds, E. Kinber, and C. Smith. On the intrinsic complexity of learning. *Information and Computation*, 123(1):64–71, 1995.

[Gol67] E. M. Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.

[HHW98] L. Hemaspaandra, H. Hempel, and G. Wechsung. Query order. *SIAM Journal of Computing*, 28:637–651, 1998.

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

[JK12] Sanjay Jain and Efim Kinber. Mind change speed-up for learning languages from positive data. In Christoph Dürr and Thomas Wilke, editors, *29th International Symposium on Theoretical Aspects of Computer Science (STACS 2012)*, volume 14 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 350–361. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012.

[JLS10] S. Jain, Q. Luo, and F. Stephan. Learnability of automatic classes. In *Language and Automata Theory and Applications, 4th International Conference, LATA 2010*, volume 6031 of *LNCS*, pages 321–332. Springer, 2010.

[JS97] S. Jain and A. Sharma. The structure of intrinsic complexity of learning. *Journal of Symbolic Logic*, 62:1187–1201, 1997.

[KN95] B. Khoussainov and A. Nerode. Automatic presentations of structures. In *Logical and Computational Complexity, (International Workshop LCC 1994)*, volume 960 of *Lecture Notes in Computer Science*, pages 367–392. Springer, 1995.

[KSW87] J Köbler, U. Schöning, and K. Wagner. The difference and truth-table hierarchies for NP. *RAIRO Theoretical Informatics and Applications*, 21:419–435, 1987.

[LZ93a] S. Lange and T. Zeugmann. Language learning in dependence on the space of hypotheses. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, pages 127–136. ACM Press, 1993.

[LZ93b] S. Lange and T. Zeugmann. Language learning with bounded number of mind changes. In *Proceedings of the Tenth Annual Symposium on Theoretical Aspects of Computer Science*, pages 682–691. Springer-Verlag, 1993. Lecture Notes Computer Science, 665.

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

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

[Wie86] R. Wiehagen. On the complexity of effective program synthesis. In K. Jantke, editor, *Analogical and Inductive Inference, Proceedings of the International Workshop*, volume 265 of *Lecture Notes in Computer Science*, pages 209–219. Springer-Verlag, 1986.

[ZZ08] T. Zeugmann and S. Zilles. Learning recursive functions: A survey. *Theoretical Computer Science A*, 397(1–3):4–56, 2008. Special Issue on Forty Years of Inductive Inference. Dedicated to the 60th Birthday of Rolf Wiehagen.