Learning from Streams

Sanjay Jain*[1], Frank Stephan**[2] and Nan Ye[3]

[1] Department of Computer Science,
National University of Singapore, Singapore 117417, Republic of Singapore.
sanjay@comp.nus.edu.sg
[2] Department of Computer Science and Department of Mathematics,
National University of Singapore, Singapore 117417, Republic of Singapore.
fstephan@comp.nus.edu.sg
[3] Department of Computer Science,
National University of Singapore, Singapore 117417, Republic of Singapore.
yenan@comp.nus.edu.sg

Abstract. Learning from streams is a process in which a group of learners separately obtain information about the target to be learned, but they can communicate with each other in order to learn the target. We are interested in machine models for learning from streams and study its learning power (as measured by the collection of learnable classes). We study how the power of learning from streams depends on the two parameters m and n, where n is the number of learners which track a single stream of input each and m is the number of learners (among the n learners) which have to find, in the limit, the right description of the target. We study for which combinations m, n and m', n' the following inclusion holds: Every class learnable from streams with parameters m, n is also learnable from streams with parameters m', n'. For the learning of uniformly recursive classes, we get a full characterization which depends only on the ratio $\frac{m}{n}$; but for general classes the picture is more complicated. Most of the noninclusions in team learning carry over to noninclusions with the same parameters in the case of learning from streams; but only few inclusions are preserved and some additional noninclusions hold. Besides this, we also relate learning from streams to various other closely related and well-studied forms of learning: iterative learning from text, learning from incomplete text and learning from noisy text.

1 Introduction

The present paper investigates the scenario where a team of learners observes data from various sources, called *streams*, so that only the combination of all these data give the complete picture of the target to be learnt; in addition the communication abilities between the team members is limited. Examples of such a scenario are the following: some scientists perform experiments to study a phenomenon, but no one has the budget to do all the necessary experiments and therefore they share the results; various earth-bound telescopes observe an object in the sky,

* Supported in part by NUS grant number R252-000-308-112.
** Supported in part by NUS grant number R146-000-114-112.

where each telescope can see the object only during some hours a day; several space ships jointly investigate a distant planet.

This concrete setting is put into the abstract framework of inductive inference as introduced by Gold [3, 6, 11]: the target to be learnt is modeled as a recursively enumerable set of natural numbers (which is called a "language"); the team of learners has to find in the limit an index for this set in a given hypothesis space. This hypothesis space might be either an indexed family or, in the most general form, just a fixed acceptable numbering of all r.e. sets. Each team member gets as input a stream whose range is a subset of the set to be learnt; but all team members together see all the elements of the set to be learnt. Communication between the team members is modeled by allowing each team member to finitely often make its data available to all the other learners. We assume that the learners communicate in the above way only finitely often.

The notion described above is denoted as $[m, n]$**StreamEx**-learning where n is the number of team members and m is the minimum number of learners out of these n which must converge to the correct hypothesis in the limit. Note that this notion of learning from streams is a variant of team learning, denoted as $[m, n]$**TeamEx**, which has been extensively studied [1, 14, 18, 19, 21, 22]; the main difference between the two notions is that in team learning, all members see the same data, while in learning from streams, each team member sees only a part of the data and can exchange with the other team members only finitely much information. In the following, **Ex** denotes the standard notion of learning in the limit from text; this notion coincides with $[1, 1]$**StreamEx** and $[1, 1]$**TeamEx**.

In related work, Baliga, Jain and Sharma [5] investigated a model of learning from various sources of inaccurate data where most of the data sources are nearly accurate.

We start with giving the formal definitions in Section 2. In Section 3 we first establish a characterization result for learning indexed families. Our main theorem in this section, Theorem 8, shows a tell-tale like characterization for learning from streams for indexed families. An indexed family $\mathcal{L} = \{L_0, L_1, \ldots\}$ is $[m, n]$**StreamEx**-learnable iff it is $[1, \lfloor \frac{n}{m} \rfloor]$**StreamEx**-learnable iff there exists a uniformly r.e. sequence $E_0, E_1, \ldots$ of finite sets such that $E_i \subseteq L_i$ and there are at most $\lfloor \frac{n}{m} \rfloor$ many languages L in $\mathcal{L}$ with $E_i \subseteq L \subseteq L_i$. Thus, for indexed families, the power of learning from streams depends only on the success ratio. Additionally, we show that for indexed families, the hierarchy for stream learning is similar to the hierarchy for team function learning (see Corollary 10); note that there is an indexed family in $[m, n]$**TeamEx** $-$ $[m, n]$**StreamEx** iff $\frac{m}{n} \leq \frac{1}{2}$. Note that these characterization results imply that the class of nonerasing pattern languages [2] is $[m, n]$**StreamEx**-learnable for all m, n with $1 \leq m \leq n$. We further show (Theorem 12) that a class $\mathcal{L}$ can be noneffectively learned from streams iff each language in $\mathcal{L}$ has a finite tell-tale set [3] with respect to the class $\mathcal{L}$, though these tell-tale sets may not be uniformly recursively enumerable from their indices. Hence the separation among different stream learning criteria is due to computational reasons rather than information theoretic reasons.

In Section 4 we consider the relationship between stream learning criteria with different parameters, for general classes of r.e. languages. Unlike the indexed family case, we show that more streaming is harmful (Theorem 14): There are classes of languages which can be learned by all n learners when the data is divided into n streams, but which cannot be learned even by one of the learners when the data is divided into $n' > n$ streams. Hence, for learning r.e. classes, $[1, n]$**StreamEx** and $[1, n']$**StreamEx** are incomparable for different $n, n' \geq 1$. This stands in contrast to the learning of indexed families where we have that $[1, n]$**StreamEx** is properly contained in $[1, n+1]$**StreamEx** for each $n \geq 1$.

Theorem 15 shows that requiring fewer number of machines to be successful gives more power to stream learning even if the success ratio is sometimes high. For each m there exists a class which is $[m, n]$**StreamEx**-learnable for all $n \geq m$ but not $[m+1, n']$**StreamEx**-learnable for any $n' \geq 2m$.

In Section 5 we first show that stream learning is a proper restriction of team learning in the sense that $[m, n]$**StreamEx** $\subset [m, n]$**TeamEx**, as long as $1 \leq m \leq n$ and $n > 1$. We also show how to carry over several separation results from team learning to learning from streams, as well as give one simulation result which carries over. In particular we show in Theorem 18 that if $\frac{m}{n} > \frac{2}{3}$ then $[m, n]$**StreamEx** $= [n, n]$**StreamEx**. Also, in Theorem 20 we show that if $\frac{m}{n} \leq \frac{2}{3}$ then $[m, n]$**StreamEx** $\not\subseteq$ **Ex**. One can similarly carry over several more separation results from team learning.

One could consider streaming of data as some form of "missing data" as each individual learner does not get to see all the data which is available, even though potentially any particular data can be made available to all the learners via synchronization. Iterative learning studies a similar phenomenon from a different perspective: though the (single) learner gets all the data, it cannot remember all of its past data; its new conjecture depends only on its just previous conjecture and the new data. We show in Theorem 21 that in the context of iterative learning, learning from streams is not restrictive (and is advantageous in some cases, as Corollary 9 can be adapted for iterative stream learners). We additionally compare stream learning with learning from incomplete or noisy data as considered in [10, 16].

2 Preliminaries and Model for Stream Learning

For any unexplained recursion theoretic notation, the reader is referred to the textbooks of Rogers [20] and Odifreddi [15]. The symbol $\mathbb{N}$ denotes the set of natural numbers, $\{0, 1, 2, 3, \ldots\}$. Subsets of $\mathbb{N}$ are referred to as languages. The symbols $\emptyset$, $\subseteq$, $\subset$, $\supseteq$ and $\supset$ denote empty set, subset, proper subset, superset and proper superset, respectively. The cardinality of a set S is denoted by $\mathrm{card}(S)$. $\max(S)$ and $\min(S)$, respectively, denote the maximum and minimum of a set S, where $\max(\emptyset) = 0$ and $\min(\emptyset) = \infty$. $\mathrm{dom}(\psi)$ and $\mathrm{ran}(\psi)$ denote the domain and range of ψ. Furthermore, $\langle \cdot, \cdot \rangle$ denotes a recursive 1–1 and onto pairing function [20] from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$ which is increasing in both its arguments: $\langle x, y \rangle = \frac{(x+y)(x+y+1)}{2} + y$. The pairing function can be extended to n-tuples by taking $\langle x_1, x_2, \ldots, x_n \rangle = \langle x_1, \langle x_2, \ldots, x_n \rangle \rangle$.

The information available to the learner is a sequence consisting of exactly the elements in the language being learned. In general, any sequence T on $\mathbb{N} \cup \{\#\}$ is called a *text*, where $\#$ indicates a pause in information presentation. $T(t)$ denotes the $(t+1)$-st element in T and $T[t]$ denotes the initial segment of T of length t. Thus $T[0] = \epsilon$, where ϵ is the empty sequence. $\mathrm{ctnt}(T)$ denotes the set of numbers in the text T. If σ is an initial segment of a text, then $\mathrm{ctnt}(\sigma)$ denotes the set of numbers in σ. Let SEQ denote the set of all initial segments. For $\sigma, \tau \in SEQ$, $\sigma \subseteq \tau$ denotes that σ is an initial segment of τ. $|\sigma|$ denotes the length of σ.

A *learner from texts* is an algorithmic mapping from SEQ to $\mathbb{N} \cup \{?\}$. Here the output ? of the learner is interpreted as "no conjecture at this time." For a learner M, one can view the sequence $M(T[0])$, $M(T[1])$, ..., as a sequence of conjectures (grammars) made by M on T.

Intuitively, successful learning is characterized by the sequence of conjectured hypotheses eventually stabilizing on correct ones. The concepts of *stabilization* and *correctness* can be formulated in various ways and we will be mainly concerned with the notion of explanatory (**Ex**) learning. The conjectures of learners are interpreted as grammars in a given hypothesis space $\mathcal{H}$, which is always recursively enumerable family of r.e. languages (in some cases, we even take the hypothesis space to be a uniformly recursive family, also called an indexed family). Unless specified otherwise, the hypothesis space is taken to be a fixed acceptable numbering $W_0, W_1, \ldots$ of all r.e. sets.

Definition 1 (Gold [11]). Given a hypothesis space $\mathcal{H} = \{H_0, H_1, \ldots\}$ and a language L, a sequence of indices $i_0, i_1, \ldots$ is said to be an **Ex**-*correct* grammar sequence for L, if there exists s such that for all $t \geq s$, $H_{i_t} = L$ and $i_t = i_s$. A learner M **Ex**-*learns* a class $\mathcal{L}$ of languages iff for every $L \in \mathcal{L}$ and every text T for L, M on T outputs an **Ex**-correct grammar sequence for L.

We use **Ex** to also denote the collection of language classes which are **Ex**-learnt by some learner.

Now we consider learning from streams. For this the learners would get streams of texts as input, rather than just one text.

Definition 2. Let $n \geq 1$. $T = (T_1, \ldots, T_n)$ is said to be a *streamed text* for L if $\mathrm{ctnt}(T_1) \cup \ldots \cup \mathrm{ctnt}(T_n) = L$. Here n is called the *degree of dispersion* of the streamed text. We sometimes call a streamed text just a *text*, when it is clear from the context what is meant.

Suppose $T = (T_1, \ldots, T_n)$ is a streamed text. Then, for all t, $\sigma = (T_1[t], \ldots, T_n[t])$, is called an *initial segment* of T. Furthermore, we define $T[t] = (T_1[t], \ldots, T_n[t])$. We define $\mathrm{ctnt}(T[t]) = \mathrm{ctnt}(T_1[t]) \cup \ldots \cup \mathrm{ctnt}(T_n[t])$ and similarly for the content of streamed texts. We let $SEQ^n = \{(\sigma_1, \sigma_2, \ldots, \sigma_n) : \sigma_1, \sigma_2, \ldots, \sigma_n \in SEQ \text{ and } |\sigma_1| = |\sigma_2| = \ldots = |\sigma_n|\}$. For $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ and $\tau = (\tau_1, \tau_2, \ldots, \tau_n)$, we say that $\sigma \subseteq \tau$ if $\sigma_i \subseteq \tau_i$ for $i \in \{1, \ldots, n\}$.

Let $\mathcal{L}$ be a language collection and $\mathcal{H}$ be a hypothesis space.

When learning from streams, a team $M_1, \ldots, M_n$ of learners accesses a streamed text $T = (T_1, \ldots, T_n)$ and works as follows. At time t, each learner M_i sees as input $T_i[t]$ plus the initial segment $T[sync_t]$, outputs a hypothesis $h_{i,t}$ and might update $sync_{t+1}$ to t. Here, initially $sync_0 = 0$ and $sync_{t+1} = sync_t$ whenever no team member updates $sync_{t+1}$ at time t.

In the following assume that $1 \leq m \leq n$. A team $(M_1, \ldots, M_n)$ $[m, n]$**StreamEx**-learns $\mathcal{L}$ iff for every $L \in \mathcal{L}$ and every streamed text T for L, (a) there is a maximal t such that $sync_{t+1} = t$ and (b) for at least m indices $i \in \{1, 2, \ldots, n\}$, the sequence of hypotheses $h_{i,0}, h_{i,1}, \ldots$ is an

Ex-correct sequence for L.

We let $[m, n]$**StreamEx** denote the collection of language classes which are $[m, n]$**StreamEx**-learnt by some team. The ratio $\frac{m}{n}$ is called the *success-ratio* of the team.

Note that a class $\mathcal{L}$ is $[1, 1]$**StreamEx**-learnable iff it is **Ex**-learnable. A further important notion is that of *team learning* [21]. This can be reformulated in our setting as follows: $\mathcal{L}$ is $[m, n]$**TeamEx**-learnable iff there is a team of learners $(M_1, \ldots, M_n)$ which $[m, n]$**StreamEx**-learn every language $L \in \mathcal{L}$ from every streamed text $(T_1, \ldots, T_n)$ for L when $T_1 = T_2 = \cdots = T_n$ (and thus each T_i is a text for L).

For notational convenience we sometimes use $M_i(T[t]) = M_i(T_1[t], \ldots, T_n[t])$ (along with $M_i(T_i[t], T[sync_t]))$ to denote M_i's output at time t when the team $M_1, \ldots, M_n$ gets the streamed text $T = (T_1, \ldots, T_n)$ as input. Note that here the learner sees several inputs rather than just one input as in the case of learning from texts (**Ex**-learning). It will be clear from context which kind of learner is meant.

One can consider updating of $sync_{t+1}$ to t as *synchronization*, as the data available to any of the learners is passed to every learner. Thus, for ease of exposition, we often just refer to updating of $sync_{t+1}$ to t by M_i as request for synchronization by M_i.

Note that in our models, there is no more synchronization after some finite time. If one allows synchronization without such a constraint, then the learners can synchronize at every step and thus there would be no difference from the team learning model. Furthermore, in our model there is no restriction on how the data is distributed among the learners. This is assumed to be done in an adversary manner, with the only constraint being that every datum appears in some stream. A stronger form would be that the data is distributed via some mechanism (for example, x, if present, is assigned to the stream $x \mod n + 1$). We will not be concerned with such distributions but only point out that learning in such a scenario is easier.

The following proposition is immediate from Definition 2.

Proposition 3. *Suppose* $1 \leq m \leq n$. *Then the following statements hold.*

- **(a)** $[m, n]$**StreamEx** $\subseteq [m, n]$**TeamEx**.
- **(b)** $[m + 1, n + 1]$**StreamEx** $\subseteq [m, n + 1]$**StreamEx**.
- **(c)** $[m + 1, n + 1]$**StreamEx** $\subseteq [m, n]$**StreamEx**.

The following definition on stabilizing sequence and locking sequences are generalizations of similar definitions for learning from texts.

Definition 4 (Based on Blum and Blum [6], Fulk [9]). Suppose that L is a language and $M_1, \ldots, M_n$ are learners. Then, $\sigma = (\sigma_1, \ldots, \sigma_n)$ is called a *stabilizing sequence* for $M_1, \ldots, M_n$ on L for $[m, n]$**StreamEx**-learning iff $\mathrm{ctnt}(\sigma) \subseteq L$ and there are at least m numbers $i \in \{1, \ldots, n\}$ such that for all streamed texts T for L with $\sigma = T[|\sigma|]$ and for all $t \geq |\sigma|$, when $M_1, \ldots, M_n$ are fed the streamed text T, for $sync_t$ and $h_{i,t}$ as defined in Definition 2, (a) $sync_t \leq |\sigma|$ and (b) $h_{i,t} = h_{i,|\sigma|}$.

A stabilizing sequence σ is called a *locking sequence* for $M_1, \ldots, M_n$ on L for $[m, n]$**StreamEx**-learning iff in (b) above $h_{i,|\sigma|}$ is additionally an index for L (in the hypothesis space used).

The following fact is based on a result of Blum and Blum [6].

Fact 5. *Assume that L is $[m,n]$**StreamEx**-learnable by $M_1,\ldots,M_n$. Then there exists a locking sequence σ for $M_1,M_2,\ldots,M_n$ on L.*

Recall that a pattern language [2] is a set of words generated from a pattern π. A pattern π is a sequence of variables and symbols (constants) from alphabet Σ. A pattern π generates a word w iff one can obtain the word w by choosing, for each variable, a value from Σ^+. We now show that the class of pattern languages is learnable from streamed text. Note that the result also follows from Theorem 8 below and the fact that pattern languages form an indexed family. We give a proof sketch below for illustrative purposes.

Example 6. *The collection of pattern languages is $[n,n]$**StreamEx**-learnable.*

Proof sketch. We construct n learners $M_1,\ldots,M_n$ which $[n,n]$**StreamEx**-learn the collection of pattern languages. On input streamed text T and at time $t+1$, M_i computes $D=\mathrm{ctnt}(T_i[t])\cup\mathrm{ctnt}(T[sync_{t+1}])$ and the learner M_i updates $sync_{t+2}$ if $T_i(t)$ is not longer than any string in D and does not belong to D. The hypothesis of M_i at time $t+1$ is the most specific pattern containing all strings in $\mathrm{ctnt}(T[sync_{t+1}])$. It is easy to see that when $t+1$ is large enough, the shortest strings in $T[sync_{t+1}]$ are just the shortest strings in the input pattern-language; thus all learners do not synchronize after that and they all output the correct pattern. □

3 Some Characterization Results

In this section we first consider a characterization for learning from streams for indexed families. Our characterization is similar in spirit to Angluin's characterization for learning indexed families.

Definition 7 (Angluin [3]). $\mathcal{L}$ is said to satisfy the *tell-tale set criterion* if for every $L\in\mathcal{L}$, there exists a finite set D_L such that for any $L'\in\mathcal{L}$ with $L'\supseteq D_L$, we have $L'\not\subset L$. D_L is called a *tell-tale set* of L. $\{D_L : L\in\mathcal{L}\}$ is called a family of tell-tale sets of $\mathcal{L}$.

Angluin [3] used the term *exact learning* to refer to learning using the language class to be learned as the hypothesis space and she showed that a uniformly recursive language class $\mathcal{L}$ is exactly **Ex**-learnable iff it has a uniformly recursively enumerable family of tell-tale sets [3]. A similar characterization holds for noneffective learning [13, pp. 42–43]: Any class $\mathcal{L}$ of r.e. languages is noneffectively **Ex**-learnable iff $\mathcal{L}$ satisfies the tell-tale criterion. For learning from streamed text, we have the following corresponding characterization.

Theorem 8. *Suppose $k\geq 1$ and $1\leq m\leq n$ and $\frac{1}{k+1}<\frac{m}{n}\leq\frac{1}{k}$. Suppose $\mathcal{L}=\{L_0,L_1,\ldots\}$ is an indexed family where one can effectively (in i,x) test whether $x\in L_i$. Then $\mathcal{L}\in[m,n]$**StreamEx** iff there exists a uniformly r.e. sequence $E_0,E_1,\ldots$ of finite sets such that for each i, $E_i\subseteq L_i$ and there are at most k sets $L\in\mathcal{L}$ with $E_i\subseteq L\subseteq L_i$.*

Proof. (⇒): Suppose $M_1, M_2, \ldots, M_n$ witness that $\mathcal{L}$ is in $[m,n]$**StreamEx**. Consider any $L_i \in \mathcal{L}$. Let $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ be a stabilizing sequence for $M_1, M_2, \ldots, M_n$ on L_i. Fix any j such that $1 \leq j \leq n$ and for all streamed texts T for L_i which extend σ, for all $t \geq |\sigma|$, $M_j(T[t]) = M_j(\sigma)$. Let $T_r = \sigma_r \#^\infty$ for $r \in \{1, \ldots, n\} - \{j\}$. Thus, for any $L \in \mathcal{L}$ and text T_j for L such that T_j extends σ_j and $\mathrm{ctnt}(\sigma) \subseteq L \subseteq L_i$, we have that m of $M_1, \ldots, M_n$ on $(T_1, \ldots, T_n)$ converge to grammars for L. Since the sequence of grammars output by M_r on $(T_1, T_2, \ldots, T_n)$ is independent of L chosen above (with the only constraint being L satisfying $\mathrm{ctnt}(\sigma) \subseteq L \subseteq L_i$), we have that there can be at most $\frac{n}{m}$ such $L \in \mathcal{L}$. Now note that a stabilizing sequence σ for $M_1, M_2, \ldots, M_n$ on L_i can be found in the limit. Let σ^s denote the s-th approximation to σ. Then one can let $E_i = \bigcup_{s \in \mathbb{N}} \mathrm{ctnt}(\sigma^s) \cap L_i$.

(⇐): Assume without loss of generality that each L_i is distinct. Let $E_{i,s}$ denote E_i enumerated within s steps by the uniform process for enumerating all the E_i's. Now, the learners $M_1, \ldots, M_n$ work as follows on a streamed text T. The learners keep variables i_t, s_t along with $sync_t$. Initially $i_0 = s_0 = 0$. At time $t \geq 0$ the learner M_j does the following: If $E_{i_t,s_t} \not\subseteq \mathrm{ctnt}(T[sync_t])$ or $E_{i_t,s_t} \neq E_{i_t,t}$ or $\mathrm{ctnt}(T_j[t]) \not\subseteq L_{i_t}$, then synchronize and let i_{t+1}, s_{t+1} be such that $\langle i_{t+1}, s_{t+1} \rangle = \langle i_t, s_t \rangle + 1$. Note that $\langle i_t, s_t \rangle$ can be recovered from $T[sync_t]$.

Note that for input streamed text T for L_i, the values of i_t, s_t converge as t goes to ∞. Otherwise, $sync_t$ also diverges, and once $sync_t$ is large enough so that $E_i \subseteq T[sync_t]$ and one considers $\langle i_t, s_t \rangle$ for which $i_t = i$ and $E_{i,s'} = E_{i,s_t}$ for $s' \geq s_t$, (note that all but finitely many values for s_t satisfy this) then the conditions above ensure that i_t, s_t and $sync_t$ do not change any further. Furthermore, $i' = \lim_{t \to \infty} i_t$ satisfies that $E_{i'} \subseteq L_i \subseteq L_{i'}$.

The output conjectures of the learners at time t are determined as follows: Let S be the set of (up to) k least elements below t such that each $j \in S$ satisfies $E_{i_t,s_t} \subseteq L_j \cap \{x : x \leq t\} \subseteq L_{i_t} \cap \{x : x \leq t\}$. Then, we allocate, for each $j \in S$, m learners to output grammars for L_j. It is easy to verify that, for large enough t, i_t and s_t would have stabilized to, say, i' and s', respectively, and S will contain every j such that $E_{i'} \subseteq L_j \subseteq L_{i'}$. Thus, the team $M_1, M_2, \ldots, M_n$ will $[m,n]$**StreamEx**-learn each L_j such that $E_{i'} \subseteq L_j \subseteq L_{i'}$ (the input language L_i is one such L_j).

The theorem follows from the above analysis. □

Here note that the direction (⇒) of the theorem holds even for arbitrary classes $\mathcal{L}$ of r.e. languages, rather than just indexed families. The direction (⇐) does not hold for arbitrary classes of r.e. languages. Furthermore, the learning algorithm given above for the direction (⇐) uses the indexed family $\mathcal{L}$ itself as the hypothesis space: so this is exact learning.

Corollary 9. *Suppose $1 \leq m \leq n, 1 \leq m' \leq n'$ and $\frac{n}{m} \geq k+1 > \frac{n'}{m'}$. Let $\mathcal{L}$ contain the following sets:*

- *the sets $\{2e + 2x : x \in \mathbb{N}\}$ for all e;*
- *the sets $\{2e + 2x : x \leq |W_e| + r\}$ for all $e \in \mathbb{N}$ and $r \in \{1, 2, \ldots, k\}$;*
- *all finite sets containing at least one odd element.*

*Then $\mathcal{L} \in [m,n]$***StreamEx** *$- [m',n']$***StreamEx** *and $\mathcal{L}$ can be chosen as an indexed family.*

Proof sketch. First we show that $\mathcal{L} \in [1, k+1]$**StreamEx**. For each e and for each $L \subseteq \{2e, 2e+2, 2e+4, \ldots\}$ with $\{2e\} \subseteq L$, let $E_L = \{2e\}$; also, for any language $L \in \mathcal{L}$ containing an odd number, let $E_L = L$. Now, for an appropriate indexing $L_0, L_1, \ldots$ of $\mathcal{L}$, $\{E_{L_i} : i \in \mathbb{N}\}$ is a collection of uniformly r.e. finite sets and for each $L \in \mathcal{L}$, there are at most $k+1$ sets $L' \in \mathcal{L}$ such that $E_L \subseteq L' \subseteq L$. Thus, $\mathcal{L} \in [1, k+1]$**StreamEx** by Theorem 8. On the other hand, for each $L \in \mathcal{L}$, one cannot effectively (in indices for L) enumerate a finite subset E_L of L such that $E_L \subseteq L' \subseteq L$ for at most k languages $L' \in \mathcal{L}$. We omit the details and the proof that $\mathcal{L}$ can be chosen as an indexed family. □

Corollary 10. *Let IND denote the collection of all indexed families. Suppose $1 \leq m \leq n$ and $1 \leq m' \leq n'$. Then $[m, n]$**StreamEx** $\cap$ IND $\subseteq [m', n']$**StreamEx** $\cap$ IND iff $\lfloor \frac{n}{m} \rfloor \leq \lfloor \frac{n'}{m'} \rfloor$.*

Remark 11. One might also study the inclusion problem for IND with respect to related criteria. One of them being conservative learning [3], where the additional requirement is that a team member M_i of a team $M_1, \ldots, M_n$ can change its hypothesis from L_d to L_e only if it has seen, either in its own stream or in the synchronized part of all streams, some datum $x \notin L_d$. If one furthermore requires that the learner is exact, that is, uses the hypothesis space given by the indexed family, then one can show that there are more breakpoints than in the case of usual team learning.

For example, there is a class which under these assumptions is conservatively $[2, 3]$**StreamEx**-learnable but not conservatively learnable. The indexed family $\mathcal{L} = \{L_0, L_1, \ldots\}$ witnessing this separation is defined as follows. Let Φ be a Blum complexity measure. For $e \in \mathbb{N}$ and $a \in \{1, 2\}$, L_{3e+a} is $\{e, e+1, e+2, \ldots\}$ if $\Phi_e(e) = \infty$ and L_{3e+a} is $\{e, e+1, e+2, \ldots\} - \{\Phi_e(e) + e + a\}$ if $\Phi_e(e) < \infty$. Furthermore, the sets $L_0, L_3, L_6, \ldots$ form a recursive enumeration of all finite sets D for which there is an e with $\Phi_e(e) < \infty$, $\min(D) = e$ and $\max(D) \in \{\Phi_e(e) + e + 1, \Phi_e(e) + e + 2\}$.

We now give learners M_1, M_2, M_3 which conservatively $[2, 3]$**StreamEx**-learn $\mathcal{L}$. On input text T, the learner M_i synchronizes at time t if

- either $\min(\mathrm{ctnt}(T_i[t])) < \min(\mathrm{ctnt}(T[sync_t]))$
- or there is an x in $\mathrm{ctnt}(T_i[t]) - \mathrm{ctnt}(T[sync_t])$ satisfying $x \leq \Phi_e(e) + 3 < 3 + \max(\mathrm{ctnt}(T_i[t]) \cup \mathrm{ctnt}(T[sync_t]))$, where $e = \min(\mathrm{ctnt}(T[sync_t]))$.

The conjectures of M_1, M_2, M_3 at time t depend only on $T[sync_t]$.

- If $\mathrm{ctnt}(T[sync_t]) = \emptyset$
  then M_1, M_2, M_3 output ?
  else let $e = \min(\mathrm{ctnt}(T[sync_t]))$ and proceed below.
- M_3 searches for d with $d \leq t \wedge L_{3d} = \mathrm{ctnt}(T[sync_t])$.
  If this d exists
  then M_3 conjectures L_{3d}
  else M_3 repeats its previous conjecture.
- Do the following for $a = 1$ and $a = 2$.
  If $\Phi_e(e) + e + a \notin \mathrm{ctnt}(T[sync_t])$
  then M_a conjectures L_{3e+a}

else if $\max(\mathrm{ctnt}(T[sync_t])) < \Phi_e(e) + e + 3$ and there is a $d \le t$ with $L_{3d} = \mathrm{ctnt}(T[sync_t])$
then M_a conjectures L_{3d}
else M_a conjectures L_{3e+3-a}.

It is left to the reader to verify the correctness and conservativeness of this learner.

To see that $\mathcal{L}$ is not conservatively learnable from a single text by a learner M using the exact hypothesis space, note that, for every e, M outputs a conjecture L_{3e+a} on some input σ^e, where $a \in \{1, 2\}$ and $\mathrm{ctnt}(\sigma^e) \subseteq \{e, e+1, \ldots\}$. Thus, there exists an e such that $\max(\mathrm{ctnt}(\sigma^e)) < \Phi_e(e)$ (otherwise, the learner could be used to solve the halting problem). Then, M would not be able to learn the set $\{e + x : x \le \Phi_e(e) + 2\} - \{e + \Phi_e(e) + a\}$ conservatively.

Note that the usage of the exact hypothesis space is essential for this remark. However, the earlier results of this section do not depend on the choice of the hypothesis space. Assume that there is a $k \in \{1, 2, 3, \ldots\}$ with $\frac{m}{n} \le \frac{1}{k} < \frac{m'}{n'}$. Then, similarly to Corollary 9, one can show that some class is conservatively $[m, n]$**StreamEx**-learnable but not conservatively $[m', n']$**StreamEx**-learnable.

The following result follows using the proof of Theorem 8 for noneffective learners. For noneffective learners one can consider every class as an indexed family. Furthermore, finitely many elements can be added to E_i to separate L_i from the finitely many subsets of it which contain E_i and are proper subsets of L_i — thus giving us a tell-tale set for L_i.

Theorem 12. *Suppose* $1 \le m \le n$. $\mathcal{L}$ *is noneffectively* $[m, n]$**StreamEx**-*learnable iff* $\mathcal{L}$ *satisfies Angluin's tell-tale set criterion.*

The above theorem shows that any separation between learning from streams with different parameters must be due to computational difficulties.

Remark 13. Behaviourally correct learning (**Bc**-learning) requires a learner to eventually output only correct hypotheses. Thus, the learner semantically converges to a correct hypothesis, but may not converge syntactically (see [8, 17] for a formal definition). Suppose $n \ge 1$. If an indexed family is $[1, n]$**StreamEx**-learnable, then it is **Bc**-learnable using an acceptable numbering as hypothesis space. This follows from the fact that an indexed family is **Bc**-learnable using an acceptable numbering as hypothesis space iff it satisfies the noneffective tell-tale criterion [4]. Hence, Gold's family [11] which consists of $\mathbb{N}$ and all finite sets is $[1, 2]$**TeamEx**-learnable but not $[1, n]$**StreamEx**-learnable for any n.

4 Relationship between various StreamEx-criteria

In this and the next section, for m, n, m', n' with $1 \le m \le n$ and $1 \le m' \le n'$, we consider the relationship between $[m, n]$**StreamEx** and $[m', n']$**StreamEx**.

We shall develop some basic theorems to show how the degree of dispersion, the success ratio and the number of successful learners required, affect the ability to learn from streams.

First, we show that the degree of dispersion plays an important role in the power of learning from streams. The next theorem shows that for any n, there are classes which are learnable from streams when the degree of dispersion is not more than n, but are not learnable from streams when the degree of dispersion is larger than n, irrespective of the success ratio.

Theorem 14. *For any $n \geq 1$, there exists a language class $\mathcal{L}$ such that $\mathcal{L} \in [n,n]$***StreamEx** $-$ $\bigcup_{n'>n}[1,n']$**StreamEx**.

Proof. Consider the class $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$, where

$$\mathcal{L}_1 = \{L : L = W_{min(L)} \wedge \forall x[\text{card}(\{(n+1)x, \ldots, (n+1)x+n\} \cap L) \leq 1]\} \text{ and}$$
$$\mathcal{L}_2 = \{L : \exists x\,[\{(n+1)x, \ldots, (n+1)x+n\} \subseteq L] \text{ and } L = W_x \text{ for the least such } x\}.$$

It is easy to verify that $\mathcal{L}$ can be $[n,n]$**StreamEx**-learnt. The learners can use synchronization to first find out the minimal element e in the input language; thereafter, they can conjecture e, until one of the learners (in its stream) observes $(n+1)x+j$ and $(n+1)x+j'$ for some x, j, j', where $j \neq j'$ and $j, j' \leq n$; in this case the learners use synchronization to find and conjecture (in the limit) the minimal x such that $\{(n+1)x, \ldots, (n+1)x+n\}$ is contained in the input language.

Now suppose by way of contradiction that $\mathcal{L}$ is $[1,n']$**StreamEx**-learnable by $M_1, \ldots, M_{n'}$ for some $n' > n$. We will use Kleene's recursion theorem to construct a language in $\mathcal{L}$ which is not $[1,n']$**StreamEx**-learned by $M_1, \ldots, M_{n'}$.

First, we give an algorithm to construct in stages a set S_e depending on a parameter e. At stage s, we construct $(\sigma_{1,s}, \ldots, \sigma_{n',s}) \in SEQ^{n'}$ where we will always have that $\sigma_{i,s} \subseteq \sigma_{i,s+1}$.

- Stage 0: $(\sigma_{1,0}, \sigma_{2,0}, \ldots, \sigma_{n',0}) = (e, \#, \ldots, \#)$. Enumerate e into S_e.
- Stage $s > 0$.

  Let $\sigma = (\sigma_{1,s-1}, \ldots, \sigma_{n',s-1})$. Search for a $\tau = (\tau_1, \ldots, \tau_{n'}) \in SEQ^{n'}$, such that (i) for $i \in \{1, \ldots, n'\}$, $\sigma_{i,s-1} \subset \tau_i$, (ii) $\min(\text{ctnt}(\tau)) = e$ and (iii) for all x, $\text{card}(\{y : y \leq n, (n+1)x + y \in \text{ctnt}(\tau)\}) \leq 1$, and one of the following holds:

  (a) One of the learners requests for synchronization after τ is given as input to the learners $M_1, \ldots, M_{n'}$.

  (b) All the learners make a mind change between seeing σ and τ, that is, for all i with $1 \leq i \leq n'$, for some τ' with $\sigma \subseteq \tau' \subseteq \tau$, $M_i(\sigma) \neq M_i(\tau')$.

  If one of the searches succeeds, then let $\sigma_{i,s} = \tau_i$, enumerate $\text{ctnt}(\tau)$ into S_e and go to stage $s+1$.

If each stage finishes, then by Kleene's recursion theorem, there exists an e such that $W_e = S_e$ and thus $W_e \in \mathcal{L}_1$. For $i \in \{1, \ldots, n'\}$, let $T_i = \bigcup_s \sigma_{i,s}$. Now, either the learners $M_1, \ldots, M_{n'}$ synchronize infinitely often or each of them makes infinitely many mind changes when the streamed text $T = (T_1, T_2, \ldots, T_{n'})$ is given to them as input. Hence $M_1, \ldots, M_{n'}$ do not $[1,n']$**StreamEx**-learn $W_e \in \mathcal{L}_1$.

Now suppose stage s starts but does not finish. Let $\sigma = (\sigma_{1,s-1}, \sigma_{2,s-1}, \ldots, \sigma_{n',s-1})$. Thus, as the learners only see their own texts and the data given to every learner up to the point of last synchronization, we have that for some j with $1 \le j \le n'$, for all $\tau = (\tau_1, \tau_2, \ldots, \tau_{n'})$ extending $\sigma = (\sigma_{1,s-1}, \sigma_{2,s-1}, \ldots, \sigma_{n',s-1})$, such that $\min(\mathrm{ctnt}(\tau)) = e$ and for all x, i, $\mathrm{card}(\{y : y \le n, (n+1)x + y \in \mathrm{ctnt}(\sigma) \cup \mathrm{ctnt}(\tau_i)\}) \le 1$, (a) none of the learners synchronize after seeing τ and (b) M_j does not make a mind change between σ and τ.

Let $rem(i) = i \mod (n+1)$. Let $x_s = 1 + \max(\mathrm{ctnt}(\sigma))$. For $1 \le i \le n'$, such that $rem(i) \neq rem(j)$, let T_i be an extension of $\sigma_{i,s}$ such that $\mathrm{ctnt}(T_i) - \mathrm{ctnt}(\sigma_{i,s}) = \{(n+1)(x_s + x) + rem(i) : x \in \mathbb{N}\}$. For $i \in \{1, \ldots, n'\}$ with $rem(i) = rem(j)$ and $i \neq j$, we let $T_i = \sigma_{i,s}\#^\infty$. We will choose T_j below such that $\sigma_{j,s-1} \subseteq T_j$ and $\mathrm{ctnt}(T_j) - \mathrm{ctnt}(\sigma_{j,s-1}) = \{(n+1)(x_s+x) + rem(j) : x_s + x \ge k\}$, for some $k > x_s$.

Let p_i be the grammar which M_i outputs in the limit, if any, when the team $M_1, \ldots, M_{n'}$ is provided with the input $(T_1, \ldots, T_{n'})$. As the learner M_i only sees T_i and the synchronized part of the streamed texts, by (a) and (b) above, we have that none of the members of team synchronize beyond σ and the learner M_j converges to the same grammar as it did after the team is provided with input σ, irrespective of which $k > x_s$ is chosen. Now, by Kleene's recursion theorem there exists a $k > x_s$ such that $W_k = \mathrm{ctnt}(\sigma_{j,s}) \cup \{(n+1)(x_s + x) + rem(j) : x_s + x \ge k\} \cup \bigcup_{i \in \{1,2,\ldots,n'\} - \{j\}} \mathrm{ctnt}(T_i)$ and $W_k \notin \{W_{p_i} : 1 \le i \le n'\}$. Hence $W_k \in \mathcal{L}_2$ and W_k is not $[1, n']$**StreamEx**-learnt by $M_1, \ldots, M_{n'}$.

The theorem follows from the above analysis. □

The following result shows that the number of successful learners affects learnability from streams crucially.

Theorem 15. *Suppose $k \ge 1$. Then, there exists an $\mathcal{L}$ such that for all $n \ge k$ and $n' \ge 2k$, $\mathcal{L} \in [k, n]$**StreamEx** but $\mathcal{L} \notin [k+1, n']$**StreamEx**.*

Proof. Let k be as in the statement of the theorem. Let ψ be a partial recursive function such that $\mathrm{ran}(\psi) \subseteq \{1, \ldots, 2k\}$, the complement of $\mathrm{dom}(\psi)$ is infinite and for any r.e. set S such that $S \cap C$ is infinite, $S \cap B$ is nonempty, where $B = \{\langle x, y \rangle : \psi(x) = y\}$ and $C = \{\langle x, j \rangle : x \notin \mathrm{dom}(\psi), 1 \le j \le 2k\}$. Note that one can construct such a ψ in a way similar to the construction of simple sets. Let $A_x = B \cup \{\langle x, j \rangle : 1 \le j \le 2k\}$. Let $\mathcal{L} = \{B\} \cup \{A_x : x \notin \mathrm{dom}(\psi)\}$.

We claim that $\mathcal{L} \in [k, n]$**StreamEx** for all $n \ge k$ but $\mathcal{L} \notin [k+1, n']$**StreamEx** for all $n' \ge 2k$.

We construct $M_1, \ldots, M_k$ which $[k, n]$**StreamEx**-learn $\mathcal{L}$ as follows.

On input $T[t] = (T_1[t], \ldots, T_n[t])$, the learners synchronize if for some i, $\mathrm{ctnt}(T_i[t-1])$ does not contain $\langle x, j \rangle$ and $\langle x, j' \rangle$ with $j \neq j'$, but $\mathrm{ctnt}(T_i[t])$ does contain such $\langle x, j \rangle$ and $\langle x, j' \rangle$.

If synchronization has happened (in some previous step), then the learners output a grammar for $B \cup \{\langle x, j \rangle : 1 \le j \le 2k\}$, where x is the unique number such that $\langle x, j \rangle$ and $\langle x, j' \rangle$ are in the synchronized text for some $j \neq j'$. Otherwise, $M_1, \ldots, M_k$ output a grammar for B and each M_i with $k+1 \le i \le n$ does the following: it first looks for the least x such that $\langle x, j \rangle \in \mathrm{ctnt}(T_i[t])$ for some j, and x is not verified to be in $\mathrm{dom}(\psi)$ in t steps; then M_i outputs a grammar for A_x if such an x is found, and outputs ? if no such x is found.

If the learners ever synchronize, then clearly all learners correctly learn the target language. Suppose no synchronization happens. If the language is B, then $M_1, \ldots, M_k$ correctly learn the input language. If the language is A_x for some $x \notin \mathrm{dom}(\psi)$, then $n \geq 2k$ (otherwise synchronization would have happened) and at least k learners among $M_{k+1}, \ldots, M_n$ eventually see exactly one pair of the form $\langle x, j \rangle$, where $1 \leq j \leq 2k$, and these learners will correctly learn the input language.

Now suppose by way of contradiction that there is a team $(M'_1, \ldots, M'_{n'})$ of learners which $[k+1, n']$**StreamEx**-learns $\mathcal{L}$. By Fact 5, there exists a locking sequence $\sigma = (\sigma_1, \ldots, \sigma_{n'})$ for the learners $M'_1, \ldots, M'_{n'}$ on B. Let $S \subseteq \{1, \ldots, n'\}$ be of size $k+1$ such that the learners M'_i, $i \in S$, do not make a mind change beyond σ on any streamed text T for B which extends σ.

By definition of ψ, there must be only finitely many $\langle x, j \rangle \in C$ such that the learners $M'_1, M'_2, \ldots, M'_{n'}$ synchronize or one of the learners M'_i, $i \in S$, makes a mind change beyond σ on any streamed text extending σ for $B \cup \{\langle x, j \rangle\}$ — otherwise we would have an infinite r.e. set S consisting of such pairs, with $S \subseteq C$ but $S \cap B = \emptyset$, a contradiction to the definitions of ψ, B, C. Let X be the set of these finitely many $\langle x, j \rangle$. Let Z be the set of x such that, for some i with $1 \leq i \leq n'$, the grammar output by M'_i on input σ is for A_x, or the grammar output by M'_i (in the limit) on input $\sigma_i \#^\infty$ (with the last point of synchronization being before all of input σ is seen) is for A_x.

Select some $z \notin \mathrm{dom}(\psi)$ such that $z \notin Z$ and $(z, j) \notin X$ for any j. Now we construct a streamed text extending σ for A_z on which the learners fail. Let $S' \supseteq S$ be a subset of $\{1, 2, \ldots, n'\}$ of size $2k$. If i is the j-th element of S' then choose T_i such that T_i extends σ_i and $\mathrm{ctnt}(T_i) = B \cup \{\langle z, j \rangle\}$ else (when $i \notin S'$) let $T_i = \sigma_i \#^\infty$. Thus, $T = (T_1, \ldots, T_{n'})$ is a streamed text for A_z. However, only the learners M'_i with $i \in S' - S$ can converge to correct grammars for A_z (as the learners M_i with $i \in S$ or $i \notin S'$, would not have converged to a grammar for A_z by definition of z, X and Z above).

It follows that $\mathcal{L} \notin [k+1, n']$**StreamEx**. □

5 Learning from Streams versus Team Learning

Team learning is a special form of learning from streams, in which all learners receive the same complete information about the underlying reality, thus team learnability provides upper bounds for learnability from streams with the same parameters. These upper bounds are strict.

Theorem 16. *Suppose* $1 \leq m \leq n$ *and* $n > 1$. *Then* $[m, n]$**StreamEx** $\subset [m, n]$**TeamEx**.

Proof. The inclusion follows from Proposition 3. The inclusion is proper as on one hand it holds that $[1, 1]$**StreamEx** $\subseteq [m, n]$**TeamEx** and on the other hand, by Theorem 14, we have $[1, 1]$**StreamEx** $\not\subseteq [m, n]$**StreamEx**. □

Remark 17. Another question is how this transfers to the learnability of indexed families. If $\frac{m}{n} > \frac{1}{2}$ and $\mathcal{L}$ is an indexed family, then $\mathcal{L} \in [m, n]$**StreamEx** iff $\mathcal{L} \in [m, n]$**TeamEx** iff $\mathcal{L} \in$ **Ex**. But if $1 \leq m \leq \frac{n}{2}$, then the class $\mathcal{L}$ consisting of $\mathbb{N}$ and all its finite subsets is $[1, 2]$**TeamEx**-learnable and $[m, n]$**TeamEx**-learnable but not $[m, n]$**StreamEx**-learnable.

Below we will show how several results from team learning can be carried over to the stream learning situation.

It was previously shown that in team learning, when the success ratios exceed a certain threshold, then the exact success ratio does not affect learnability any longer. Using a similar *majority* argument, we can show similar collapsing results for learning from streams (Theorem 18 and Theorem 19).

Before we formulate this precisely, we introduce two useful concepts. First, by s-m-n theorem, there exists a recursive function *majority* such that $majority(g_1, \ldots, g_n)$ is a grammar for $\{x : x$ is a member of more than half of $W_{g_0}, \ldots, W_{g_n}\}$. Note that if more than half of $g_1, \ldots, g_n$ are grammars for a language L, then $majority(g_1, \ldots, g_n)$ is a grammar for L as well.

Second, suppose $M_1, \ldots, M_n$ are a team learning from a given streamed text $T = (T_1, \ldots, T_n)$. Then we can define the convergence time $Conv_T(i, t)$ at time t for M_i to be the minimum $t' \geq 0$ such that whenever $t' \leq j \leq t$, $M_i(T_1[j], \ldots, T_n[j]) = M_i(T_1[t'], \ldots, T_n[t'])$. Thus a necessary condition for M_i to learn the target (in **Ex**-sense) is that $\lim_{t\to\infty} Conv_T(i, t)$ converges.

Theorem 18. *Suppose* $1 \leq m \leq n$. *If* $\frac{m}{n} > \frac{2}{3}$, *then* $[m, n]\mathbf{StreamEx} = [n, n]\mathbf{StreamEx}$.

Proof. We construct $M'_1, \ldots, M'_n$ such that they $[n, n]\mathbf{StreamEx}$-learn $\mathcal{L}$. The basic idea of the proof is that the learners $M'_1, M'_2, \ldots, M'_n$ maintain the convergence information for the seemingly earliest m converging machines among $M_1, \ldots, M_n$ (breaking ties in favour of lower numbered learner) based on the input seen so far. If this information gets corrupted (due to one of the m earliest converging learners among $M_1, \ldots, M_n$ making a mind change), then synchronization is used to update the information.

Suppose $T = (T_1, \ldots, T_n)$ is the input streamed text for a language L. Initially, $sync_0 = 0$. Each learner, at time $t \geq 1$, has information about $Conv_T(i, sync_t)$ for each i. At time $t \geq 1$, each learner first computes $i_1^t, i_2^t, \ldots, i_n^t$ as a permutation of $1, 2, \ldots, n$ such that, for r with $1 \leq r < n$, $Conv_T(i_r^t, sync_t) \leq Conv_T(i_{r+1}^t, sync_t)$ and if $Conv_T(i_r^t, sync_t) = Conv_T(i_{r+1}^t, sync_t)$, then $i_r^t < i_{r+1}^t$. Now the learner M'_i synchronizes at time t if either M_i synchronizes at time t or $i = i_r^t$ for some r with $1 \leq r \leq m$ and $M_i(T_i[t], T[sync_t]) \neq M_i(T_i[t-1], T[sync_{t-1}])$ (recall that M_i sees only the information in $T_i[t]$ and $T[sync_t]$ at time t). The grammar output by M'_i is $majority(g_1, g_2, \ldots, g_m)$, where $g_r = M_{i_r^t}(T[sync_t])$.

It is easy to verify that if the learners $M_1, M_2, \ldots, M_n$, $[m, n]\mathbf{StreamEx}$-learn L, then eventually (as t goes to ∞) $sync_t$ and the variables $i_1^t, \ldots, i_m^t$ get stabilized and the learners $M_{i_1^t}, \ldots, M_{i_m^t}$ would have converged to their final grammar after having seen the input $T[sync_t]$ and $T_{i_1^t}[t], \ldots,$ $T_{i_m^t}[t]$, respectively. Thus, $majority(g_1, g_2, \ldots, g_m)$ would be a correct grammar for L as at least $m - (n - m)$ of the grammars $g_1, \ldots, g_m$ are correct grammars for L. □

Theorem 19. *Suppose* $1 \leq m \leq n$ *and* $k \geq 1$. *Then* $[\lfloor \frac{2k}{3} \rfloor (n - m) + km, kn]\mathbf{StreamEx} \subseteq [m, n]\mathbf{StreamEx}$.

One can also carry over several diagonalization results from team learning to learning from streams. An example is the following.

Theorem 20. *For all* $j \in \mathbb{N}$, $[j + 2, 2j + 3]\mathbf{StreamEx} \not\subseteq [j + 1, 2j + 1]\mathbf{TeamEx}$.

Proof. Let $\mathcal{L}_j = \{L : \mathrm{card}(L) \geq j+3$ and if $e_0 < \ldots < e_{j+2}$ are the $j+3$ smallest elements of L, then either $[W_{e_0} = \ldots = W_{e_{j+1}} = L]$ or [at least one of $e_0, \ldots, e_{j+1}$ is a grammar for L and $W_{e_{j+2}}$ is finite and $max(W_{e_{j+2}})$ is a grammar for $L]\}$.

$\mathcal{L}$ is clearly in $[j+2, 2j+3]$**StreamEx**, as the learners can first obtain the least $j+3$ elements in the input texts (via synchronization, whenever a smaller element than previous $j+3$ smallest elements is observed). Then, $j+2$ learners could just output $e_0, e_1, \ldots, e_{j+1}$ and the remaining learners output (in the limit) $max(W_{e_{j+2}})$, if it exists.

The proof to show that $\mathcal{L}_j \notin [j+1, 2j+1]$**TeamEx** can be done essentially using the technique of [12]. Below we give the proof for the case of $j = 0$. Thus, we need to show that $\mathcal{L}_0 \notin$ **Ex**. Suppose $\mathcal{L}_0$ is **Ex**-learnable by a learner M. We give an algorithm using a recursive function p as parameter to construct a sequence of uniformly r.e. sets $S_0, S_1, S_2, \ldots$, in stages.

- At stage 0, $\sigma_0 = p(0)p(1)p(2)$ and enumerate $p(0), p(1), p(2)$ into S_0 and S_1.
- At stage $s > 0$, let x_s be the minimum element such that no $x \geq x_s$ is enumerated into S_0 or S_1. Enumerate $p(x_s)$ into S_0 and S_2 and enumerate all elements of S_0 into S_{x_s}. Now dovetail between the searches in (a) and (b) below:
  (a) Search for $p(x_s)$ in an enumeration of $W_{M(\sigma_{s-1})}$.
  (b) Search for τ with $\mathrm{ctnt}(\tau)$ consisting of numbers greater than $p(2)$ such that $M(\sigma_{s-1}\tau) \neq M(\sigma)$.

  If the search in (a) succeeds first, then enumerate $p(x_s+1)$ into S_1 and S_2 and enumerate all elements in S_1 into S_{x_s+1}. Continue the search in (b).

  Whenever the search in (b) succeeds, let $\sigma_s = \sigma_{s-1}\tau$ and let $S = S_0 \cup S_1 \cup \mathrm{ctnt}(\sigma_s)$. Enumerate elements in S into S_0 and S_1. Go to Stage $s+1$.

The construction of $S_0, S_1, \ldots$ is effective in p, thus there exists a recursive function f_p such that $W_{f_p(i)} = S_i$. By operator recursion theorem [7], there exists a monotone increasing recursive p such that $f_p = p$. Fix this p. The way we add elements into S_0 and S_1 guarantees that $p(0) < p(1) < p(2)$ are the smallest elements in $W_{p(0)}$ and $W_{p(1)}$.

If the construction goes through infinitely many stages, then the search in (b) is always successful and $W_{p(0)} = W_{p(1)} = L$ for some L. Thus $L \in \mathcal{L}_0$. However, $\bigcup_{i\in\mathbb{N}} \sigma_i$ is a text for L and M makes infinitely many mind changes on it.

If some stage s starts but does not terminate, then M does not change its mind no matter how σ_{s-1} is extended by using numbers greater than $p(2)$. If the search in (a) is not successful, then $W_{p(0)} = W_{p(x_s)} = L$ for some L and $p(x_s)$ is the maximum element in $W_{p(2)}$. Thus $L \in \mathcal{L}_0$. Extend σ_{s-1} to be a text for L. However, in this case M on this text has stabilized on $M(\sigma_{s-1})$, but the language $W_{M(\sigma_{s-1})}$ is not equal to L as $p(x_s)$ is in L but not in $W_{M(\sigma_{s-1})}$.

If the search in (a) is successful, then $W_{p(1)} = W_{p(x_s+1)} = L$ for some L and $p(x_s+1)$ is the maximum element in $W_{p(2)}$. Thus $L \in \mathcal{L}_0$. Extend σ_{s-1} to be a text for L. However, in this case M on this text has stabilized on $M(\sigma_{s-1})$, but the language $W_{M(\sigma_{s-1})}$ is not equal to L as $p(x_s)$ is in $W_{M(\sigma_{s-1})}$ but not in L.

Hence $\mathcal{L}_0 \notin$ **Ex**. □

6 Iterative Learning and Learning from Inaccurate Texts

In this section, the notion of learning from streams is compared with other notions of learning where the data is used by the learner in more restricted ways or the data is presented in more adversarial manner than in the standard case of learning. The first notion to be dealt with is iterative learning where the learner only remembers the most recent hypothesis, but does not remember any past data [23]. Later, we will consider other adversary input forms: for example the case of incomplete texts where finitely many data-items might be omitted [10, 16] or noisy texts where finitely many data-items (not in the input language) might be added to the input text.

The motivation for iterative learning is the following: When humans learn, they do not memorize all past observed data, but mainly use the hypothesis they currently hold, together with new observations to formulate new hypotheses. Many scientific results can be considered to be obtained in iterative fashion. Iterative learning for learning from a single stream/text was previously modeled by requiring the learners to be a function of the previous hypothesis and the current observed data. Formally, a single-stream learner $M : (\mathbb{N} \cup \{\#\})^* \to (\mathbb{N} \cup \{?\})$ is *iterative* if there exists a recursive function $F : (\mathbb{N} \cup \{?\}) \times (\mathbb{N} \cup \{\#\}) \to \mathbb{N} \cup \{?\}$ such that on a text T, $M(T[0]) =?$ and for $t > 0$, $M(T[t]) = F(M(T[t-1]), T(t))$. For notational simplicity, we shall write $F(M(T[t-1]), T(t))$ as $M(M(T[t-1]), T(t))$. We can similarly define iterative learning from several streams by requiring each learner's hypothesis to be a recursive function of its previous hypothesis and the set of the newest datum received by each learner — here, when synchronization happens, the learners only share the latest data seen by the learners rather than the whole history of data seen.

Iterative learning can be considered as a form of information incompleteness as the learner(s) do not memorize all the past observed data. Interestingly, every iteratively learnable class is learnable from streams irrespective of the parameters.

Theorem 21. *For any $n \geq 1$, every language class* **Ex***-learnable by an iterative learner is iteratively $[n, n]$***StreamEx***-learnable.*

Proof. Suppose $\mathcal{L}$ is **Ex**-learnable by an iterative learner M. We construct $M_1, \ldots, M_n$ which $[n, n]$**StreamEx**-learn $\mathcal{L}$. We maintain the invariant that each M_i outputs the same grammar g at each time step. Initially $g =?$. At any time t, suppose M_i receives a datum x_i^t, previous hypothesis is g and the synchronized data, if any, was $d_1^t, d_2^t, \ldots, d_n^t$. The output conjecture of the learners is $g' = g$, if there is no synchronized data; otherwise the output conjecture of the learners is $g' = M(\ldots M(M(g, d_1^t)d_2^t) \ldots d_n^t)$. The learner M_i requests for synchronization if $M(g', x_i^t) \neq g'$. Clearly $M_1, \ldots, M_n$ form a team of iterative learners from streams and always output the same hypothesis. Furthermore, it can be seen that if M on the text $T_1(0)T_2(0) \ldots T_n(0)T_1(1)T_2(1) \ldots T_n(1) \ldots$ converges to a hypothesis, then the sequence of hypothesis output by learners $M_1, M_2, \ldots, M_n$ also converges to the same hypothesis. Thus, if M iteratively learns the input language, then $M_1, M_2, \ldots, M_n$ also iteratively $[n, n]$**StreamEx**-learn the input language. □

Now we compare learning from streams with learning from an incomplete or noisy text. Formally, a text $T \in (\mathbb{N} \cup \{\#\})^\infty$ is an *incomplete* text for L iff $L \supseteq \mathrm{ctnt}(T)$ and $L - \mathrm{ctnt}(T)$ is finite [10, 16]. A text for L is noisy iff $\mathrm{ctnt}(T) \subseteq L$ and $\mathrm{ctnt}(T) - L$ is finite [16]. **Ex**-learning from incomplete or noisy texts is the same as **Ex**-learning except that the texts are now incomplete texts or noisy texts, respectively. In the following we investigate the relationships of these criteria with learning from streams. We show that learning from streams is incomparable to learning from incomplete or noisy texts.

The nature of information incompleteness in learning from an incomplete text is very different from the incompleteness caused by streaming of data, because streaming only spreads information, but does not destroy information (Theorem 12), while the incompleteness in an incomplete text involves the destruction of information. This difference is made precise by the following incomparability results.

Proposition 22. *Suppose that $\mathcal{L}$ consists of $L_0 = \mathbb{N}$ and all sets $L_{k+1} = \{1+\langle x, y\rangle : x \leq k \wedge y \in \mathbb{N}\}$. Then $\mathcal{L} \in [n, n]$**StreamEx** for any $n \geq 1$ but $\mathcal{L}$ can neither be **Ex**-learnt from noisy text nor from incomplete text. Furthermore, $\mathcal{L}$ is iteratively learnable.*

Proof. $\mathcal{L}$ is iteratively learnable by the following algorithm: as long as 0 has not been seen in the input, the learner conjectures L_{k+1} for the minimal number k such that no element $1+\langle x, y\rangle$ with $x > k$ and $y \in \mathbb{N}$ has been seen so far; once 0 has been observed, the learner changes its mind to L_0 and does no further mind change. It follows that $\mathcal{L}$ is iteratively $[m, n]$**StreamEx**-learnable.

For the negative result, note that the presence of 0 in the text distinguishes the learning of L_0 from that of L_{k+1}, $k \geq 0$. However, by either omitting 0 from the text of L_0 in the case of learning from incomplete texts or by adding it to the text of any L_{k+1} in the case of noisy texts, this method of distinguishing the two cases gets lost and the resulting situation is similar to Gold's example that $\mathbb{N}$ and the sets $\{0, 1, 2, \ldots, x\}$ form an unlearnable class [11]. $\square$

For the separations in the converse direction, one cannot use indexed families as every indexed family **Ex**-learnable from normal text is already learnable from streams; obviously this implication survives when learnability from normal text is replaced by learnability from incomplete or noisy text.

Remark 23. Suppose $n \geq 2$. Then the cylindrification of the class $\mathcal{L}$ from Theorem 14 is **Ex**-learnable from incomplete text but not $[1, n]$**StreamEx**-learnable.

Here the cylindrification of the class $\mathcal{L}$ is just the class of all sets $\{\langle x, y\rangle : x \in L \wedge y \in \mathbb{N}\}$ with $L \in \mathcal{L}$. Incomplete texts for a cylindrification of such a set L can be translated into standard texts for L and so the learnability from incomplete texts can be established; the diagonalization against the stream learners carries over.

It is known that learnability from noisy text is possible only if for every two different sets L, L' in the class the differences $L - L'$ and $L' - L$ are both infinite. This is a characterization for the case of indexed families, but it is only a necessary but not sufficient criterion for classes in general. For example if a class $\mathcal{L}$ consists of sets $L_x = \{\langle x, y\rangle : y \in \mathbb{N} - \{a_x\}\}$ without any method to obtain a_x from x in the limit, then learnability from noisy text is lost.

Theorem 24. *There is a class $\mathcal{L}$ which is learnable from noisy text but not $[1, n]$**StreamEx**-learnable for any $n \geq 2$.*

In the following only the separating class is given. The class $\mathcal{L}$ is the set of all sets L such that there exist d, e such that L satisfies one of the following two conditions:

- φ_d is defined on some finite domain, φ_e extends φ_d, $e > \max(\mathrm{dom}(\varphi_d))$, φ_e is total and L contains $\langle x, y, z\rangle$ iff $x = 0 \wedge y = d$ or $x > 0 \wedge \varphi_e(x-1) = y$ or $x = e+1 \wedge y = \varphi_e(e) + 1$.
- φ_d has an infinite domain and L contains $\langle x, y, z\rangle$ iff $x = 0 \wedge y = d$ or $x > 0 \wedge \varphi_d(x-1)\downarrow = y$.

So the set L is the cylindrificated graph of a partial multivalued function f for which $f(0)$ gives away the index d and the position of a double value (if it exists) gives away the index e. This class $\mathcal{L}$ is then learnable from noisy text but not $[m, n]$**StreamEx**-learnable.

7 Conclusion

In this paper we investigated learning from several streams of data. For learning indexed families, we characterized the classes which are $[m, n]$**StreamEx**-learnable using a tell-tale like characterization: An indexed family $\mathcal{L} = \{L_0, L_1, \ldots\}$ is $[m, n]$**StreamEx**-learnable iff it is $[1, \lfloor\frac{n}{m}\rfloor]$**StreamEx**-learnable iff there exists a uniformly r.e. sequence $E_0, E_1, \ldots$ of finite sets such that $E_i \subseteq L_i$ and there are at most $\lfloor\frac{n}{m}\rfloor$ many languages L in $\mathcal{L}$ such that $E_i \subseteq L \subseteq L_i$.

For general classes of r.e. languages, our investigation shows that the power of learning from streams depends crucially on the degree of dispersion, the success ratio and the number of successful learners required. Though higher degree of dispersion is more restrictive in general, we show that any class of languages which is iteratively learnable is also iteratively learnable from streams even if one requires all the learners to be successful. There are several open problems and our results suggest that there may not be a simple way to complete the picture of relationship between various $[m, n]$**StreamEx** learning criteria.

References

1. Andris Ambainis. Probabilistic inductive inference: a survey. *Theoretical Computer Science*, 264:155–167, 2001.
2. Dana Angluin. Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, 21:46–62, 1980.
3. Dana Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45:117–135, 1980.
4. Ganesh Baliga, John Case and Sanjay Jain. The synthesis of language learners. *Information and Computation*, 152:16–43, 1999.

5. Ganesh Baliga, Sanjay Jain and Arun Sharma. Learning from multiple sources of inaccurate data. *SIAM Journal on Computing*, 26:961–990, 1997.
6. Lenore Blum and Manuel Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28:125–155, 1975.
7. John Case. Periodicity in generations of automata. *Mathematical Systems Theory*, 8:15–32, 1974.
8. John Case and Christopher 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.
9. Mark Fulk. Prudence and other conditions on formal language learning. *Information and Computation*, 85:1–11, 1990.
10. Mark Fulk and Sanjay Jain. Learning in the presence of inaccurate information. *Theoretical Computer Science*, 161:235–261, 1996.
11. E. Mark Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.
12. Sanjay Jain and Arun Sharma. Computational limits on team identification of languages. *Information and Computation*, 130:19–60, 1996.
13. Sanjay Jain, Daniel Osherson, James S. Royer and Arun Sharma. *Systems That Learn: An Introduction to Learning Theory, 2nd edition. MIT Press*, 1999.
14. Sanjay Jain and Arun Sharma. Team learning of computable languages. *Theory of Computing Systems*, 33:35–58, 2000.
15. Piergiorgio Odifreddi. *Classical Recursion Theory.* North-Holland, Amsterdam, 1989.
16. Daniel Osherson, Michael Stob and Scott Weinstein. *Systems That Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists. MIT Press*, 1986.
17. Daniel Osherson and Scott Weinstein. Criteria of language learning. *Information and Control*, 52:123–138, 1982.
18. Leonard Pitt. Probabilistic inductive inference. *Journal of the ACM*, 36:383-433, 1989.
19. Leonard Pitt and Carl H. Smith. Probability and plurality for aggregations of learning machines. *Information and Computation*, 77:77-92, 1988.
20. Hartley Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, New York, 1967. Reprinted, MIT Press 1987.
21. Carl H. Smith. The power of pluralism for automatic program synthesis. *Journal of the ACM*, 29:1144–1165, 1982.
22. Carl H. Smith. Three decades of team learning. *Algorithmic Learning Theory* 1994, Springer LNCS 872:211–228, 1994.
23. Rolf Wiehagen. Limes-Erkennung rekursiver Funktionen durch spezielle Strategien. *Elektronische Informationsverbarbeitung und Kybernetik* (*EIK*), 12:93–99, 1976.