Available online at www.sciencedirect.com

SCIENCE DIRECT®

Information and Computation 204 (2006) 989–1011

Information and Computation

www.elsevier.com/locate/ic

Mind change efficient learning

Wei Luo, Oliver Schulte *

School of Computing Science, Simon Fraser University, Vancouver, Canada

Received 12 September 2005; revised 17 February 2006
Available online 2 May 2006

Abstract

This paper studies efficient learning with respect to mind changes. Our starting point is the idea that a learner that is efficient with respect to mind changes minimizes mind changes not only globally in the entire learning problem, but also locally in subproblems after receiving some evidence. Formalizing this idea leads to the notion of *strong mind change optimality*. We characterize the structure of language classes that can be identified with at most α mind changes by some learner (not necessarily effective): a language class $\mathcal{L}$ is identifiable with α mind changes iff the accumulation order of $\mathcal{L}$ is at most α. Accumulation order is a classic concept from point-set topology. We show that accumulation order is related to other established notions of structural complexity, such as thickness and intrinsic complexity. To aid the construction of learning algorithms, we show that the characteristic property of strongly mind change optimal learners is that they output conjectures (languages) with maximal accumulation order. We illustrate the theory by describing strongly mind change optimal learners for various problems such as identifying linear subspaces, one-variable patterns, and fixed-length patterns.

1991 MSC: 68Q32
Keywords: Inductive inference; Mind change complexity; Accumulation order

* Corresponding author.
E-mail addresses: wluoa@cs.sfu.ca (W. Luo); oschulte@cs.sfu.ca (O. Schulte).

0890-5401/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.ic.2006.02.004

1. Introduction

One of the goals of computational learning theory is to design learning algorithms for which we can provide performance guarantees. Identification in the limit is a central performance goal in Gold's language learning paradigm [11]. A well-studied refinement of this notion is *identification with bounded mind changes* [9,1]. In this paper, we investigate a further refinement that we term strong mind change optimality (SMC-optimality). Briefly, a learner is SMC-optimal if the learner achieves the best possible mind change bound not only for the entire problem, but also relative to any data sequences that the learner may observe.

The general theory in this paper has two main goals: (1) To provide necessary and sufficient conditions for a language collection to be identifiable with a given (ordinal) mind-change bound by some learner (not necessarily effective). (2) To provide necessary and sufficient conditions for a learner to be SMC-optimal. The results addressing (1) help us determine when an SMC-optimal learning algorithm exists, and the results addressing (2) help us to construct optimal learning algorithms when they do exist.

We situate our study in the framework of point-set topology. Previous work has shown the usefulness of topology for learning theory [34,chapter 10,28,17,4,29] We show how to view a language collection as a topological space; this allows us to apply Cantor's classic concept of *accumulation order* which assigns an ordinal $\mathrm{acc}(\mathcal{L})$ to a language collection, if $\mathcal{L}$ has bounded accumulation order. We show that a language collection $\mathcal{L}$ is identifiable with mind change bound α by a learner if and only if $\mathrm{acc}(\mathcal{L}) = \alpha$. This result establishes a purely information-theoretic and structural necessary condition for identification with bounded mind changes. Based on the concept of accumulation order, we provide necessary and sufficient conditions for a learner to be SMC-optimal. These results show that SMC-optimality strongly constrains the conjectures of learners. We illustrate these results by analyzing various learning problems, such as identifying a linear subspace and a one-variable pattern.

The paper is organized as follows. Section 2 reviews standard concepts for language identification and presents our definition of mind change optimality. Then we establish the correspondence between mind change complexity and accumulation order. Section 4 gives necessary and sufficient conditions for a learner to be strongly mind change optimal. Next, we describe some general principles for constructing SMC-optimal effective learners and illustrate them with one-variable and fixed-length pattern languages. In Section 6, we show strong relationships between the concept of accumulation order and other structural notions studied in learning theory, such as thickness [41], elasticity [44,30], and intrinsic complexity [10,13].

2. Preliminaries: language identification

2.1. Standard concepts

We employ notation and terminology from [14,27,chapter 1,11]. We write $\mathbb{N}$ for the set of natural numbers: $\{0, 1, 2, \ldots\}$. The symbols $\subseteq, \supseteq, \subset, \supset$, and $\emptyset$, respectively, stand for subset, superset, proper subset, proper superset, and the empty set. We view a language as a set of strings. We identify strings with natural numbers encoding them. Thus, we define a **language** to be a subset

of $\mathbb{N}$ and write L for a generic language [11, p. 449]. A **language learning problem** is a collection of languages; we write $\mathcal{L}$ for a generic collection of languages. A **text** T is a mapping of $\mathbb{N}$ into $\mathbb{N} \cup \{\#\}$, where # is a symbol not in $\mathbb{N}$. (The symbol # models pauses in data presentation.) We write content(T) for the intersection of $\mathbb{N}$ and the range of T. A text T is **for** a language L iff $L = \text{content(T)}$. The initial sequence of text T of length n is denoted by $T[n]$. The set of all finite initial sequences over $\mathbb{N} \cup \{\#\}$ is denoted by SEQ. We also use $SEQ(\mathcal{L})$ to denote finite initial sequences consistent with languages in $\mathcal{L}$. We let σ and τ range over SEQ. We write content(σ) for the intersection of $\mathbb{N}$ and the range of σ. The initial sequence of σ of length n is denoted by $\sigma[n]$. We say that a language L is **consistent** with σ iff content(σ) $\subseteq$ L. We write $\sigma \subset T$ or $T \supset \sigma$ to denote that text T extends initial sequence σ.

Examples.

(1) Let $L_i \equiv \{n : n \geqslant i\}$, where $i \in \mathbb{N}$; we use COINIT to denote the class of languages $\{L - i : i \in \mathbb{N}\}$ [1, p. 324].
(2) In the n-dimensional linear space $\mathbb{Q}^n$ over the field of rationals $\mathbb{Q}$, we can effectively encode every vector $\vec{v}$ by a natural number. Then a linear subspace of $\mathbb{Q}^n$ corresponds to a language. We write LINEAR_n for the collection of all (encodings of) linear subspaces of $\mathbb{Q}^n$.

A **learner** is a function that maps a finite sequence to a language or the question mark ?, meaning "no answer for now." We normally use the Greek letter Ψ and variants to denote a learner. Our term "learner" corresponds to the term "scientist" in [27, chapter 2.1.2]. In typical applications, we have available a syntactic representation for each member of the language collection $\mathcal{L}$ under investigation. In such settings, we assume the existence of an index for each member of $\mathcal{L}$, that is, a function $index : \mathcal{L} \mapsto \mathbb{N}$ (cf. [12, p. 18]), and we can take a **learning function** to be a function that maps a finite sequence to an index for a language (learning functions are called "scientists" in [12, chapter 3.3]). A computable learning function is a **learning algorithm**. We use the general notion of a learner for more generality and simplicity until we consider issues of computability.

Let $\mathcal{L}$ be a collection of languages. A learner Ψ **for** $\mathcal{L}$ is a mapping of SEQ into $\mathcal{L} \cup \{?\}$. Thus the learners we consider are class-preserving; for the results in this paper, this assumption carries no loss of generality. Usually context fixes the language collection $\mathcal{L}$ for a learner Ψ.

We say that a learner Ψ **identifies** a language L on a text T for L, if $\Psi(T[n]) = L$ for all but a finite number of stages n. Next we define identification of a language collection relative to some evidence.

Definition 2.1. A learner Ψ identifies $\mathcal{L}$ given $\sigma \iff$ for every language $L \in \mathcal{L}$, and for every text $T \supset \sigma$ for L, we have that Ψ identifies L on T.

Thus, a learner Ψ identifies a language collection $\mathcal{L}$ if Ψ identifies $\mathcal{L}$ given the empty sequence Λ.

Examples.

(1) The following learner Ψ_{CO} identifies COINIT: If content(σ) $= \emptyset$, then $\Psi_{\text{CO}}(\sigma) := ?$. Otherwise set $m := \min(\text{content}(\sigma))$, and set $\Psi_{\text{CO}}(\sigma) := L_m$.
(2) Let vectors(σ) be the set of vectors whose code numbers appear in σ. Then define $\Psi_{\text{LIN}}(\sigma) = \text{span}(\text{vectors}(\sigma))$, where span($V$) is the linear span of a set of vectors V. The learner Ψ_{LIN} identifies LINEAR_n. The problem of identifying a linear subspace of reactions arises in particle

physics, where it corresponds to the problem of finding a set of conservation principles governing observed particle reactions [22,42]. Interestingly, it appears that the theories accepted by the particle physics community match the output of Ψ_{LIN} [43,39,40].

A learner Ψ **changes its mind** at some nonempty finite sequence $\sigma \in \mathrm{SEQ}$ if $\Psi(\sigma) \neq \Psi(\sigma^-)$ and $\Psi(\sigma^-) \neq ?$, where σ^- is the initial segment of σ with σ's last element removed [10,1]. (No mind changes occur at the empty sequence Λ.)

Definition 2.2 (*based on [1]*)**.** Let Ψ be a learner and c be a function that assigns an ordinal to each finite sequence $\sigma \in \mathrm{SEQ}$.

(1) c is a **mind-change counter** for Ψ and $\mathcal{L}$ if $c(\sigma) < c(\sigma^-)$ whenever Ψ changes its mind at some nonempty sequence σ. When $\mathcal{L}$ is fixed by context, we simply say that c is a mind change counter for Ψ.
(2) Ψ identifies a class of languages $\mathcal{L}$ **with mind-change bound** α given $\sigma \iff \Psi$ identifies $\mathcal{L}$ given σ and there is a mind-change counter c for Ψ and $\mathcal{L}$ such that $c(\sigma) = \alpha$.
(3) A language collection $\mathcal{L}$ is **identifiable with mind change bound** α given $\sigma \iff$ there is a learner Ψ such that Ψ identifies $\mathcal{L}$ with mind change bound α given σ.

Examples.

(1) For COINIT, define a counter c_0 as follows: $c_0(\sigma) := \omega$ if $\mathrm{content}(\sigma) = \emptyset$, where ω is the first transfinite ordinal, and $c_0(\sigma) := \min(\mathrm{content}(\sigma))$ otherwise. Then c_0 is a mind change counter for Ψ_{CO} given Λ. Hence Ψ_{CO} identifies COINIT with mind change bound ω (cf. [1, Section 1]).
(2) For LINEAR_n, define the counter $c_1(\sigma)$ by $c_1(\sigma) := n - \dim(\mathrm{span}(\mathrm{vectors}(\sigma)))$, where $\dim(V)$ is the dimension of a space V. Then c_1 is a mind change counter for Ψ_{LIN} given Λ, so Ψ_{LIN} identifies LINEAR_n with mind change bound n.
(3) Let FIN be the class of languages $\{D \subseteq \mathbb{N} : D \text{ is finite}\}$. Then a learner that always conjectures $\mathrm{content}(\sigma)$ identifies FIN. However, there is no mind change bound for FIN [1].

2.2. Strong mind change optimality

In this section, we introduce a new identification criterion that is the focus of this paper. Our point of departure is the idea that learners that are efficient with respect to mind changes should minimize mind changes not only globally in the entire learning problem but also locally after receiving specific evidence. For example, in the COINIT problem, the best global mind change bound for the entire problem is ω [1, Section 1], but after observing initial data $\langle 5 \rangle$, a mind change efficient learner should succeed with at most 5 more mind changes, as does Ψ_{CO}. However, there are many learners that require more than 5 mind changes after observing $\langle 5 \rangle$ yet still succeed with the optimal mind change bound of ω in the entire problem.

To formalize this motivation, consider a language collection $\mathcal{L}$. If a mind change bound exists for $\mathcal{L}$ given σ, we write $\mathrm{MC}_{\mathcal{L}}(\sigma)$ for the least ordinal α such that $\mathcal{L}$ is identifiable with α mind changes given σ. We require that a learner should succeed with $\mathrm{MC}_{\mathcal{L}}(\sigma)$ mind changes after each

data sequence $\sigma \in \mathrm{SEQ}(\mathcal{L})$. For example, the learner Ψ_{CO} achieves this performance for COINIT. This leads us to the following definition.

Definition 2.3. A learner Ψ is **strongly mind change optimal** for $\mathcal{L}$ if there is a mind change counter c for Ψ such that $c(\sigma) = \mathrm{MC}_{\mathcal{L}}(\sigma)$ for all sequences σ.

We use the abbreviation "SMC-optimal" for "strongly mind change optimal" (The terminology and intuition is similar to Kelly's in [19,21]). A learner Ψ is simply SMC-optimal for $\mathcal{L}$ if Ψ is SMC-optimal given Λ.

Examples.

(1) In the COINIT problem, $\mathrm{MC}_{\mathcal{L}}(\Lambda) = \omega$, and $\mathrm{MC}_{\mathcal{L}}(\sigma) = \min(\mathrm{content}(\sigma))$ when $\mathrm{content}(\sigma) \neq \emptyset$. Since c_0 is a mind change counter for Ψ_{CO}, it follows that Ψ_{CO} is SMC-optimal. Any learner Ψ such that (1) $\Psi(\sigma) = \Psi_{\mathrm{CO}}(\sigma)$ if $\mathrm{content}(\sigma) \neq \emptyset$ and (2) $\Psi(\sigma) = \Psi(\sigma^-)$ if $\mathrm{content}(\sigma) = \emptyset$ is also SMC-optimal. (The initial conjecture $\Psi(\Lambda)$ is not constrained.)
(2) The learner Ψ_{LIN} is SMC-optimal. Thus for the problem of inferring conservation laws, *SMC-optimality coincides with the inferences of the physics community*.

Discussion. In our paper [25], we examined a weaker notion of mind change efficiency termed "uniform mind change optimality." The difference with strong mind change optimality is that uniform mind change optimality requires the mind change counter to take on the lowest value (i.e., $c(\sigma) = \mathrm{MC}_{\mathcal{L}}(\sigma)$) only when the learner produces an output consistent with the data (i.e., when $\Psi(\sigma)$ is consistent with σ). Formally, a learner Ψ is **uniformly mind change optimal** for $\mathcal{L}$ if there is a mind change counter c for Ψ such that for all sequences σ, if $\Psi(\sigma) \neq ?$ and $\Psi(\sigma)$ is consistent with σ, then $c(\sigma) = \mathrm{MC}_{\mathcal{L}}(\sigma)$. For noneffective learners, strong mind change optimality is no more stringent than uniform mind change optimality: we will show in Theorem 3.1 that if a mind change bound of α is feasible for the learning problem $\mathcal{L}$, then there is a strongly mind change optimal learner that realizes the bound α. The two notions of optimality differ, however, for computable learners: because consistency with the data may be hard to achieve for a computable learner, uniform mind change optimality can be attained by a computable learner in more problems than strong mind change optimality. We will describe an example separating the two notions in Section 4 after analyzing the properties of strongly mind change optimal learners. Both notions are useful in the study of learning algorithms. The stricter notion of strong mind change optimality, the topic of the current paper, is mathematically more straightforward than uniform mind change optimality. As the examples in this paper show, even though this requirement is more stringent in general for computable learners, it can be met effectively in a number of natural learning problems, such as identifying a linear subspace and a one-variable pattern (Section 5).

3. A topological characterization of mind-change bounded identifiability

Information-theoretical aspects of inductive inference have been studied by many learning theorists (e.g., [12,27]). As Jain et al. observe [12, p. 34]:

> Many results in the theory of inductive inference do not depend upon computability assumptions; rather, they are information theoretic in character. Consideration of noncomputable scientists thereby facilitates the analysis of proofs, making it clearer which assumptions carry the burden.

As an example, Angluin showed that her Condition 1 characterizes the indexed families of nonempty recursive languages inferable from positive data by computable learners [3,p. 121] and that the noneffective version, Condition 2, is a necessary condition for inferability by computable learners.[2] Variants of Angluin's Condition 2 turn out to be both sufficient and necessary for various models of language identifiability by noncomputable learners ([27, Chapter 2.2.2], [12, Theorem 3.26]). Information theoretic requirements such as Condition 2 constitute necessary conditions for computable learners, and are typically the easiest way to prove the unsolvability of some learning problems when they do apply. For example, Apsitis used the Baire topology on total recursive functions to show that $\mathbf{EX}_\alpha \neq \mathbf{EX}_{\alpha+1}$ [4, Section 3]. On the positive side, if a sufficient condition for noneffective learnability is met, it often yields insights that lead to the design of a successful learning algorithm.

It has often been observed that point-set topology, one of the most fundamental and well-studied mathematical subjects, provides useful concepts for describing the information theoretic structure of learning problems [34, chapter 10,28,4,17,29]. In particular, Apsitis investigated the mind change complexity of function learning problems in terms of the Baire topology [4]. He showed that Cantor's 1883 notion of accumulation order in a topological space [8] defines a natural ordinal-valued measure of complexity for function learning problems, and that accumulation order provides a lower bound on the mind change complexity of a function learning problem. We generalize Apsitis' use of topology to apply it to language collections. The following section briefly reviews the relevant topological concepts.

3.1. Basic definitions in point-set topology

A **topological space** over a set X is a pair $(X, \mathcal{O})$, where $\mathcal{O}$ is a collection of subsets of X, called **open sets**, such that $\emptyset$ and X are in $\mathcal{O}$ and $\mathcal{O}$ is closed under arbitrary union and finite intersection. One way to define a topology for a set is to find a base for it. A **base** $\mathcal{B}$ for X is a class of subsets of X such that

(1) $\bigcup \mathcal{B} = X$, and
(2) for every $x \in X$ and any $B_1, B_2 \in \mathcal{B}$ that contain x, there exists $B_3 \in \mathcal{B}$ such that $x \in B_3 \subseteq B_1 \cap B_2$.

For any base $\mathcal{B}$, the set $\{\bigcup \mathcal{C} : \mathcal{C} \subseteq \mathcal{B}\}$ is a topology for X [23, p. 52]. That is, we may take an open set to be a union of sets in the base. Let $\mathcal{L}$ be a class of languages and $\sigma \in \mathrm{SEQ}$. We use $\mathcal{L}|\sigma$ to denote all languages in $\mathcal{L}$ that are consistent with σ (i.e., $\{L \in \mathcal{L} : L \text{ is consistent with } \sigma\}$); similarly $\mathcal{L}|D$ denotes the languages in $\mathcal{L}$ that include a given finite subset D. The next proposition shows that $\mathcal{B}_\mathcal{L} = \{\mathcal{L}|\sigma : \sigma \in \mathrm{SEQ}\}$ constitutes a base for $\mathcal{L}$.

[2] Condition 2 characterizes BC-learnability for computable learners [6].

Proposition 3.1. *$\mathcal{B}_{\mathcal{L}} = \{\mathcal{L}|\sigma : \sigma \in SEQ\}$ is a base for $\mathcal{L}$; hence $\mathcal{T}_{\mathcal{L}} = \{\bigcup \mathcal{S} : \mathcal{S} \subseteq \mathcal{B}_{\mathcal{L}}\}$ is a topology for $\mathcal{L}$.*

The topology $\mathcal{T}_{\mathcal{L}}$ generalizes the positive information topology from recursion theory [33, p. 186] if we consider the graphs of functions as languages (as in [12, chapter 3.9.2][27, chapter 2.6.2]).

Examples. For the language collection COINIT we have that $\text{COINIT}|\{2,3\} = \{L_0, L_1, L_2\}$ and $\text{COINIT}|\{0\} = \{L_0\}$. The base $\mathcal{B}_{\text{COINIT}}$ consists of all sets of the form $\text{COINIT}|d$, where d is a finite subset of $\mathbb{N}$.

In a topological space $(X, \mathcal{T})$, a point x is an **isolated point** of a set $A \subseteq X$ if there is an open set $O \in \mathcal{T}$ such that $x \in O$ and $A \cap O \setminus \{x\} = \emptyset$. If x is not an isolated point of $A \subseteq X$, then x is an **accumulation point of** A. Following Cantor [8], we define the **derived sets** using the concept of accumulation points.

Definition 3.1 (*Cantor*)**.** Let $(X, \mathcal{T})$ be a topological space.

(1) The 0**-th derived set** of X, denoted by $X^{(0)}$, is just X.
(2) For every successor ordinal α, the α**-th derived set** of X, denoted by $X^{(\alpha)}$, is the set of all accumulation points of $X^{(\alpha-1)}$.
(3) For every limit ordinal α, the set $X^{(\alpha)}$ is the intersection of all β-th derived sets, where $\beta < \alpha$. That is, $X^{(\alpha)} = \bigcap_{\beta<\alpha} X^{(\beta)}$.

We give an example from the topology of the real plane that illustrates the geometrical intuitions behind the topological concepts.

Example. Let

$$A = \left\{\left(\frac{1}{n}, \frac{1}{m}\right) : n, m \in \mathbb{N}\right\} \cup \left\{\left(\frac{1}{n}, 0\right) : n \in \mathbb{N}\right\} \cup \left\{\left(0, \frac{1}{m}\right) : m \in \mathbb{N}\right\}$$

be a set of points in the real plane $\mathbb{R}^2$ with the standard topology. We use $iso(X)$ to denote all isolated points in X. Then $iso(A) = \{(\frac{1}{n}, \frac{1}{m}) : n, m \in \mathbb{N}\}$. Therefore

$$A^{(1)} = \left\{\left(\frac{1}{n}, 0\right) : n \in \mathbb{N}\right\} \cup \left\{\left(0, \frac{1}{m}\right) : m \in \mathbb{N}\right\}.$$

Similarly, we have $A^{(2)} = (0, 0)$, and $A^{(3)} = \emptyset$ (see Fig. 1).

In the topology $\mathcal{T}_{\mathcal{L}}$, a language L is an isolated point of $\mathcal{L}$ iff there is a finite subset $D \subseteq L$ such that the observation of D entails L (i.e., $\mathcal{L}|D = \{L\}$). The derived sets of $\mathcal{L}$ can be defined inductively as shown in Defnition 3.1. Note if $\alpha < \beta$ then $\mathcal{L}^{(\alpha)} \supseteq \mathcal{L}^{(\beta)}$. It can be shown in set theory that there is an ordinal α such that $\mathcal{L}^{(\beta)} = \mathcal{L}^{(\alpha)}$, for all $\beta > \alpha$ [16]. In other words, there must be a fixpoint for the derivation operation. If $\mathcal{L}$ has an empty fixpoint, then we say $\mathcal{L}$ is **scattered** [23, p.78]. In a non-scattered space, the nonempty fixed point is called a **perfect kernel**.

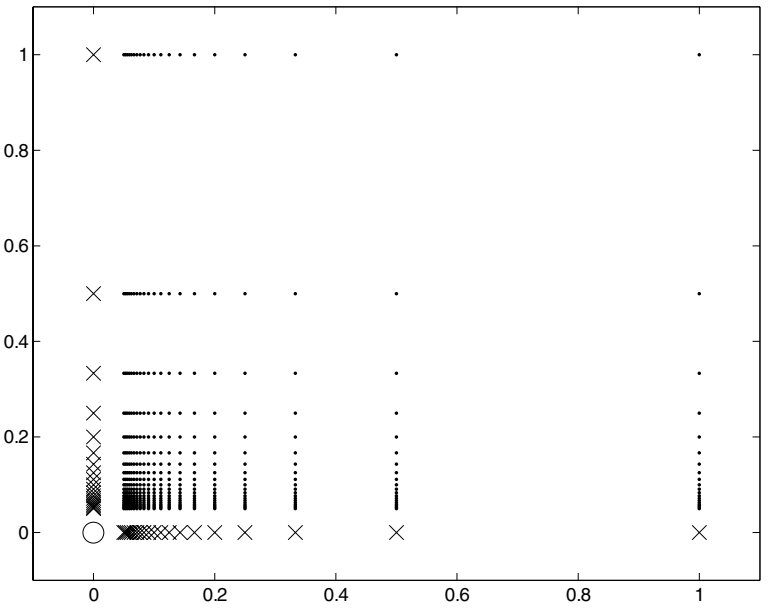

Fig. 1. A set A on the real plane. Applying derivation once will remove the points marked with dots; applying derivation twice will remove the points marked with crosses; applying derivation again will remove the point marked with the circle.

The **accumulation order of a language** L in $\mathcal{L}$, denoted by $\mathrm{acc}_{\mathcal{L}}(\mathrm{L})$ is the maximum ordinal α such that $L \in \mathcal{L}^{(\alpha)}$; when $\mathcal{L}$ is fixed by context, we simply write $\mathrm{acc}(\mathrm{L}) = \alpha$. The **accumulation order of a class of languages** $\mathcal{L}$, denoted by $\mathrm{acc}(\mathcal{L})$, is the supremum of the accumulation order of all languages in it. Therefore, a language collection has an accumulation order if and only if it is scattered.[3]

Examples.

(1) The only isolated point in COINIT is $L_0 = \mathbb{N}$, for $\mathrm{COINIT}|\{0\} = \{L_0\}$. Therefore $\mathrm{COINIT}^{(1)} = \{L_i : i \geqslant 1\}$. Similarly L_1 is the only isolated point in $\mathrm{COINIT}^{(1)}$; hence $\mathrm{COINIT}^{(2)} = \{L_i : i \geqslant 2\}$. It is easy to verify that $\mathrm{COINIT}^{(n)} = \{L_i : i \geqslant n\}$. Therefore, the accumulation order of language L_i in COINIT is i and the accumulation order of COINIT is $\omega = \sup \mathbb{N}$.

(2) In $\mathrm{LINEAR}_n = \{\text{linear subspaces of } \mathbb{Q}^n\}$, the only isolated point is $\mathbb{Q}^n$ itself: Let S be a set of n linearly independent points in $\mathbb{Q}^n$; then $\mathrm{LINEAR}_n|S = \{\mathbb{Q}^n\}$. Similarly every $(n-i)$-dimensional linear subspace of $\mathbb{Q}^n$ is an isolated point in $\mathrm{LINEAR}_{\mathrm{n}}^{(\mathrm{i})}$. Therefore, the accumulation order of LINEAR_n is n.

(3) In FIN, there is *no* isolated point. This is because for every finite subset S of $\mathbb{N}$, there are infinitely many languages in FIN that are consistent with S. Therefore, FIN is a perfect kernel of itself and FIN has no accumulation order.

[3] Accumulation order is also called scattering height, derived length, Cantor-Bendixson rank, or Cantor-Bendixson length [16].

3.2. Accumulation order characterizes mind change complexity

In this section, we show that the accumulation order of a language collection $\mathcal{L}$ is an exact measure of its mind change complexity for (not necessarily effective) learners: if $\mathrm{acc}(\mathcal{L})$ is unbounded, then $\mathcal{L}$ is not identifiable with any ordinal mind change bound; and if $\mathrm{acc}(\mathcal{L}) = \alpha$, then $\mathcal{L}$ is identifiable with a mind change bound.[4]

In a language topology, accumulation order has two fundamental properties that we apply often. Let $\mathrm{acc}_{\mathcal{L}}(\sigma) \equiv \sup\{\mathrm{acc}_{\mathcal{L}}(\mathrm{L}) : \mathrm{L} \in \mathcal{L}|\sigma\}$; as usual, we omit the subscript in context. A language L **tops** $\mathcal{L}|\sigma$ if $\mathrm{acc}_{\mathcal{L}}(\mathrm{L}) = \mathrm{acc}_{\mathcal{L}}(\sigma)$; a sequence σ is topped if there is some language that tops σ. Note that if $\mathrm{acc}_{\mathcal{L}}(\sigma)$ is a successor ordinal (e.g., finite), then σ is topped. All data sequences in $\mathrm{SEQ}(\mathrm{LINEAR}_n)$ are topped. In COINIT, the initial sequence Λ is *not* topped. A language L **uniquely tops** σ in $\mathcal{L}$ if L is the only language that tops σ in $\mathcal{L}$.

Lemma 3.1. *Let $\mathcal{L}$ be a scattered class of languages with bounded accumulation order.*

(1) *For every language $L \in \mathcal{L}$, for every text T for L, there exists a time n such that L uniquely tops $T[n]$ in $\mathcal{L}$; moreover, for every $m > n$, language L uniquely tops $T[m]$ in $\mathcal{L}$.*
(2) *For any two languages $L_1, L_2 \in \mathcal{L}$ such that $L_1 \subset L_2$ it holds that* $\mathrm{acc}_{\mathcal{L}}(\mathrm{L}_1) > \mathrm{acc}_{\mathcal{L}}(\mathrm{L}_2)$.

Proof. Part 2 is immediate. Part 1: For contradiction, assume there is a text T for L such that for all n, $\mathcal{L}|(T[n])$ contains some language L' such that $\mathrm{acc}(\mathrm{L}') \geqslant \mathrm{acc}(\mathrm{L}) = \alpha$. Then L is an accumulation point of $\mathcal{L}^{(\alpha)}$, the subclass of $\mathcal{L}$ that contains all languages with accumulation order less than or equal to α. Therefore $\mathrm{acc}(\mathrm{L}) \geqslant \alpha + 1$, which is a contradiction. □

We now establish the correspondence between mind change complexity and accumulation order: $\mathrm{MC}_{\mathcal{L}}(\sigma) = \mathrm{acc}_{\mathcal{L}}(\sigma)$.

Theorem 3.1. *Let $\mathcal{L}$ be a language collection and let σ be a finite data sequence. Then there is a learner Ψ that identifies $\mathcal{L}$ given σ with mind change bound α* $\iff \mathrm{acc}_{\mathcal{L}}(\sigma) \leqslant \alpha$.

Proof. ($\Leftarrow$) Define the mind change counter c by $c(\sigma) := \mathrm{acc}_{\mathcal{L}}(\sigma)$. We show that c is a mind change counter for the following learner Ψ that identifies $\mathcal{L}$:

(1) $\Psi(\Lambda) :=?$,
(2) $\Psi(\sigma) :=?$ if $\mathrm{acc}_{\mathcal{L}}(\sigma) < \mathrm{acc}_{\mathcal{L}}(\sigma^-)$,

[4] Necessary and sufficient conditions for finite mind change identifiability by learning *algorithms* appear in [24,32]. An anonymous referee provided the following example of a learning problem whose information-theoretic mind change complexity is 0, but that requires 1 mind change for any computable learner. Let φ be an acceptable programming system and Φ be a complexity measure for φ (see [12, chapter 2]). Let $\mathcal{L}_1$ contain all total recursive functions f such that (1) $\varphi_{f(0)} = f$, and (2) $\Phi_{f(0)}(x) \geqslant f(x+1)$ for all $x \in \mathbb{N}$. The functions in $\mathcal{L}_1$ are "self-describing" [12, Definition 4.24]. Let $\mathcal{L}_2$ contain all total recursive functions g such that (1) there exists an $x \in \mathbb{N}$ such that $\Phi_{g(0)}(x) \geqslant g(x+1)$ implies $\varphi_{g(0)}(x) \neq g(x)$ and (2) $\varphi_{g(x_0+2)} = g$ where x_0 is the least such x. The problem $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$ is identifiable with zero mind change by a noncomputable learner by waiting for the datum specifying the value of the target function at 0. On the other hand, every computable learner for $\mathcal{L}$ makes one mind change in the worst case.

(3) $\Psi(\sigma) := L$ if $\mathrm{acc}_{\mathcal{L}}(\sigma) = \mathrm{acc}_{\mathcal{L}}(\sigma^-)$ and L uniquely tops σ in $\mathcal{L}$,
(4) $\Psi(\sigma) := \Psi(\sigma^-)$ if $\mathrm{acc}_{\mathcal{L}}(\sigma) = \mathrm{acc}_{\mathcal{L}}(\sigma^-)$ and there is no language L that uniquely tops σ in $\mathcal{L}$.

It is easy to see that Ψ identifies $\mathcal{L}$. Let T be any text for any language $L \in \mathcal{L}$. Then by Lemma 3.1(1) there is a time m such that L uniquely tops $\mathcal{L}|T[n']$ for all $n' > m$. Hence Clause 3 applies at all times $n' > m$, and Ψ converges to L on T, as required.

It remains to show that c is a mind change counter for Ψ. We begin with an auxiliary observation (a): For all languages $L \notin \mathcal{L}|\sigma$, if $\Psi(\sigma) \neq L$, then $\Psi(\tau) \neq L$ for every $\tau \in \mathrm{SEQ}(\mathcal{L})$ such that $\tau \supset \sigma$. In other words, if Ψ rejects a hypothesis L inconsistent with σ, then Ψ never returns to L after σ. To see that this holds, consider some $\tau \supset \sigma$ and suppose for reductio that $\Psi(\tau) = L$. Then there must be some ν with $\sigma \subset \nu \subseteq \tau$ such that $\Psi(\nu^-) \neq L$ and $\Psi(\nu) = L$. Then Clause 3 implies that L uniquely tops ν, which contradicts the assumption that L is inconsistent with σ and hence with ν.

We argue that (*) if Clause 3 applies at σ, then no mind change occurs at σ, such that either $\Psi(\sigma) = \Psi(\sigma^-)$ or $\Psi(\sigma^-) = ?$. Suppose that $\Psi(\sigma^-) = L' \neq L$ and $\mathrm{acc}_{\mathcal{L}}(\sigma) = \mathrm{acc}_{\mathcal{L}}(\sigma^-)$ and L uniquely tops σ. Let $n < |\sigma|$ be the least time such that $\Psi(\sigma[n]) = L'$. Then by definition of Ψ, Clause 3 applies at $\sigma[n]$, and so L' uniquely tops $\sigma[n]$ in $\mathcal{L}$. Since L uniquely tops σ and $L \neq L'$, we know that $\mathrm{acc}_{\mathcal{L}}(\sigma) < \mathrm{acc}_{\mathcal{L}}(\sigma[\mathrm{n}])$, and therefore $n < |\sigma^-|$ since $\mathrm{acc}_{\mathcal{L}}(\sigma) = \mathrm{acc}_{\mathcal{L}}(\sigma^-)$.

Thus, $\mathrm{acc}_{\mathcal{L}}(\sigma^-) < \mathrm{acc}_{\mathcal{L}}(\sigma[\mathrm{n}])$. Therefore by Clause 2, there is some time m such that $n < m < |\sigma|$ such that $\Psi(\sigma[m]) = ?$, and moreover $L' \notin \mathcal{L}|\sigma[m]$. Therefore the observation (a) implies that $\Psi(\sigma^-) \neq L'$. This contradiction shows that either $\Psi(\sigma^-) = ?$ or $\Psi(\sigma^-) = L$, and thus no mind change occurs at σ, as required.

It is immediate from the construction that Ψ changes its mind at σ only if Clauses 2 or 3 apply, so (*) implies that Ψ changes its mind only if Clause 2 applies. In that case $c(\sigma) = \mathrm{acc}_{\mathcal{L}}(\sigma) < \mathrm{acc}_{\mathcal{L}}(\sigma^-) = \mathrm{c}(\sigma^-)$. So counter c is a mind change counter for Ψ since this holds for all mind changes of Ψ.

($\Rightarrow$) Let Ψ be a learner that identifies $\mathcal{L}$ given σ. Suppose c is a mind change counter such that $c(\sigma) = \alpha$. We prove by transfinite induction that if $\mathrm{acc}(\sigma) > \alpha$, then c is not a mind change counter for $\mathcal{L}$. Assume the claim holds for all $\beta < \alpha$ and consider α. Suppose $\mathrm{acc}(\sigma) > \alpha$; then there is $L \in \mathcal{L}|\sigma$ such that $\mathrm{acc(L)} = \alpha + 1$. Case 1: $\Psi(\sigma) = L$. Then since L is a limit point of $\mathcal{L}^{(\alpha)}$, there is L' in $\mathcal{L}^{(\alpha)}$ such that $L' \neq L$ and $\mathrm{acc(L')} = \alpha$. Let $T' \supset \sigma$ be a text for L'. Since Ψ identifies L', there is a time $n > |\sigma|$ such that $\Psi(T'[n]) = L'$. Since $\Psi(T'[n]) \neq \Psi(\sigma)$ and $\Psi(\sigma) \neq ?$, this is a mind change of Ψ, hence $c(T'[n]) < c(\sigma)$. That is, $c(T'[n]) = \beta < \alpha$. On the other hand, since $\mathrm{acc(L')} = \alpha$, we have $\mathrm{acc(T'[n])} > \beta$. By inductive hypothesis, c is not a mind change counter for Ψ. Case 2: $\Psi(\sigma) \neq L$. Let $T \supset \sigma$ be a text for L. Since Ψ identifies L, there is a time $n > |\sigma|$ such that $\Psi(T[n]) = L$. As $c(T[n]) \leqslant c(\sigma) = \alpha$ and $\mathrm{acc(T[n])} > \alpha$, as in Case 1, c is not a mind change counter for Ψ. □

Corollary 3.1. *Let $\mathcal{L}$ be a class of languages. Then there exists a mind-change bound for $\mathcal{L}$ if and only if $\mathcal{L}$ is scattered in the topology $\mathcal{T}_{\mathcal{L}}$.*

4. Necessary and sufficient conditions for strongly mind change optimal learners

Theorem 3.1 establishes that if the accumulation order of a language collection $\mathcal{L}$ is bounded by an ordinal α, then there is a learner Ψ that identifies $\mathcal{L}$ with at most α mind changes; moreover, the proof of the theorem shows that there is a strongly mind-change optimal learner Ψ that does so.

The goal of this section is to characterize the behavior of strongly mind-change optimal learners. These results allow us to design mind change optimal learners and to prove their optimality.

Proposition 4.1. *Let Ψ be a learner that identifies a language collection $\mathcal{L}$. Then Ψ is SMC-optimal for $\mathcal{L}$ if and only if for all data sequences σ:*

(1) *If there is a language L topping σ in $\mathcal{L}$, and $\Psi(\sigma) \neq ?$, then $\Psi(\sigma)$ uniquely tops σ in $\mathcal{L}$.*
(2) *If $\sigma \neq \Lambda$ is not topped and $\mathrm{acc}_{\mathcal{L}}(\sigma) = \mathrm{acc}_{\mathcal{L}}(\sigma^-)$, then no mind change occurs at σ.*

Proof. ($\Rightarrow$) Clause 2 follows immediately from the fact that if Ψ is SMC-optimal, then $\mathrm{acc}_{\mathcal{L}}$ is a mind change counter for Ψ. For Clause 1, suppose σ is topped. Assume for contradiction that $\Psi(\sigma) = L' \neq ?$ and L' is not the only language topping $\mathcal{L}|\sigma$. Then there exists a language $L \in \mathcal{L}|\sigma$ such that $L \neq L'$ and $\mathrm{acc}_{\mathcal{L}}(\mathrm{L}) = \mathrm{acc}_{\mathcal{L}}(\sigma)$. Let T be a text for L such that $T \supseteq \sigma$. If Ψ identifies $\mathcal{L}$, there exists a time $n > |\sigma|$ such that $\Psi(T[n]) = L$. Therefore Ψ makes at least one mind change between σ and $T[n]$. If $\mathrm{acc}_{\mathcal{L}}$ is a mind change counter for Ψ, then $\mathrm{acc}_{\mathcal{L}}(\sigma) > \mathrm{acc}_{\mathcal{L}}(\mathrm{T[n]})$. On the other hand, we have $\mathrm{acc}_{\mathcal{L}}(\mathrm{T[n]}) = \mathrm{acc}_{\mathcal{L}}(\mathrm{L}) = \mathrm{acc}_{\mathcal{L}}(\sigma)$. This contradiction shows that Ψ is not SMC-optimal.

($\Leftarrow$) We want to show that $\mathrm{acc}_{\mathcal{L}}$ is a mind change counter for Ψ.

Let σ be an arbitrary sequence in $\mathrm{SEQ}(\mathcal{L})$. There are four cases to consider:

(1) σ is topped and $\mathrm{acc}(\sigma) < \mathrm{acc}(\sigma^-)$.
(2) σ is topped and $\mathrm{acc}(\sigma) = \mathrm{acc}(\sigma^-)$.
(3) σ is not topped and $\mathrm{acc}(\sigma) < \mathrm{acc}(\sigma^-)$.
(4) σ is not topped and $\mathrm{acc}(\sigma) = \mathrm{acc}(\sigma^-)$.

We argue that $\Psi(\sigma) \neq \Psi(\sigma^-)$ and $\Psi(\sigma^-) \neq ?$ imply $\mathrm{acc}(\sigma) < \mathrm{acc}_{\mathcal{L}}(\sigma^-)$ in all four cases. That is, if a mind change occurs at σ, then the accumulation order drops at σ.

In cases 1 and 3, the implication holds trivially. In case 4, we have by Condition 2 of the proposition that there is no mind change at σ.

Case 2a: $\Psi(\sigma^-) = ?$; then there is no mind change at σ. Case 2b: $\Psi(\sigma^-) \neq ?$. We note that σ^- is topped since $\mathrm{acc}(\sigma) = \mathrm{acc}(\sigma^-)$ and σ is topped. So $\Psi(\sigma^-)$ has the highest accumulation order by Condition 1. Since $\Psi(\sigma^-)$ and $\Psi(\sigma)$ both have the highest accumulation order $\mathrm{acc}(\sigma)$, we have $\Psi(\sigma^-) = \Psi(\sigma)$. □

Proposition 4.1 shows that the key property of strongly mind change optimal learners is that when they output a consistent informative conjecture L different from ?, the conjecture L maximizes accumulation order. In many applications, hypotheses with higher accumulation order are intuitively simpler than those with lower accumulation order. In such language collections, we can think of mind change optimal methods as choosing the simplest hypothesis consistent with the data when a unique simplest hypothesis is available.[5]

[5] We are indebted to S. Jain for suggesting this interpretation of Proposition 4.1. Kelly develops the idea of linking mind change efficient learning with simplicity of hypotheses, and presents it as a formalization of Occam's Razor [21,20].

4.1. Strong vs. uniform mind change optimality

Both UMC-optimal and SMC-optimal learners share the key property of Proposition 4.1 (see [25, Section 4] for a characterization of UMC-optimal learners). The main difference is that if an SMC-optimal learner Ψ conjectures a language L on data σ^- and L is inconsistent with subsequent data σ such that σ is topped by some language L', then Ψ must change its mind at σ, whereas a UMC-optimal learner may "hang on" to a refuted hypothesis. Thus the conjectures of SMC-optimal learners must be consistent with the data σ whenever σ is topped (taking ? to be trivially consistent with any data). The proof of Theorem 3.1 shows that this consistency requirement is not restrictive for general uncomputable learners. The matter is different for effective learners: There are learning problems for which there is a computable UMC-optimal learner but no computable SMC-optimal learner [18].[6] Thus for computable learners, SMC-optimality defines a new class of learning problems.

5. Effective strongly mind change optimal learning

In this section, we consider further computational issues and illustrate how our analysis of mind change complexity can aid the design of mind change efficient learning algorithms in specific problems. As it turns out, Angluin's well-known pattern languages bring out a number of general points about constructing SMC-optimal learning algorithms.

It is straightforward to computationally implement the learners Ψ_{CO} and Ψ_{LIN}. These learners have the feature that whenever they produce a conjecture L on data σ, the language L is a subset of every other languages in $\mathcal{L}|\sigma$. Formally, we say L is the $\subseteq$**-minimum** at σ if L is a subset of every other language in $\mathcal{L}|\sigma$. It follows from Clause 2 of Lemma 3.1 that a $\subseteq$-minimum also maximizes accumulation order, so Ψ_{CO} and Ψ_{LIN} always output the language uniquely having the highest accumulation order and hence by Proposition 4.1 they are both SMC-optimal. For a language collection $\mathcal{L}$ like COINIT and LINEAR, if we can compute the $\subseteq$-minimum, an SMC-optimal learning algorithm for $\mathcal{L}$ can be constructed on the model of Ψ_{CO} and Ψ_{LIN}. However, these conditions are much stronger than necessary in general. In general, it suffices that we can *eventually* compute a $\subseteq$-minimum along any text. In particular, we can make a learner output ? when it is computationally impossible or too complex to find a $\subseteq$-minimum consistent language. We illustrate this point by specifying SMC-optimal learning algorithms for P_1 and T_n, two subclasses of languages defined by Angluin's well-known patterns [2, p. 48].

[6] Kelly and Schulte provide an example showing that the difference between UMC and SMC-optimality in fact allows for a vast gap in the computational abilities of the learners. They describe a learning problem such that (1) the problem can be solved with 1 mind change by a computable learner that is uniformly mind change optimal, but (2) no strongly mind change optimal computable learner can identify the right answer in the limit, even when augmented with an oracle for all problems of arithmetic (sets in the arithmetical hierarchy) [18]. In this example, there is a computable learner that achieves the optimal information-theoretic mind change bound of 1, but no consistent computable learner that does so. The anonymous referee's example in footnote 4 also illustrates how consistency can prevent a computable learner from achieving a mind change bound of 1, although in that example every computable learner requires more than 0 mind changes, which is the information-theoretic complexity of the problem; hence no computable learner is UMC-optimal or SMC-optimal.

5.1. Patterns

Let X be a set of variable symbols (e.g., $x_1, x_2, \ldots$) and let Σ be a finite alphabet of at least two constant symbols (e.g., $0, 1, \ldots, n$). A **pattern**, denoted by p, q etc., is a finite non-null sequence over $X \cup \Sigma$ (e.g., $\mathtt{x}_1\mathtt{0}\mathtt{x}_1$ or $\mathtt{x}_1\mathtt{x}_2\mathtt{x}_2$). We use $var(p)$ to denote the set of distinct variables in p and use $\#var(p)$ to denote the number of distinct variables in p. A pattern p is **canonical** if $var(p) = \{x_1, x_2, \ldots, x_{\#var(p)}\}$ and their first occurrence (from left to right) is in that order. For example, the pattern $\mathtt{x}_1\mathtt{2}\mathtt{x}_2\mathtt{x}_1$ is canonical, but patterns $\mathtt{x}_2\mathtt{1}\mathtt{x}_4$ and $\mathtt{x}_2\mathtt{x}_1$ are not. We use PATTERN to denote the set of all canonical patterns. A **substitution** θ replaces a variable in a pattern p by another pattern uniformly. For example, $\theta = [\mathtt{x}_2\mathtt{x}_3/x_1]$ maps the pattern $\mathtt{x}_1\mathtt{x}_1$ to the pattern $\mathtt{x}_2\mathtt{x}_3\mathtt{x}_2\mathtt{x}_3$. Substitutions give rise to a partial order over all patterns. Following [37,38], we say that a pattern q **subsumes** a pattern p, denoted by $p \preceq q$, if there is a substitution θ such that $p = q\theta$. The **language generated by a pattern** p, denoted by $L(p)$, is the set $\{q \in \Sigma^* : q \preceq p\}$. The **length** of a pattern p, denoted by $|p|$, is the number of symbols occuring in p. The set of strings of the same length as a given pattern p plays an important role in the proofs below; we denote this set by $S(p) \equiv \{s \in L(p) : |s| = |p|\}$. We observe that for an alphabet Σ, the size of $S(p)$ is given by $|S(p)| = |\Sigma|^{\#var(p)}$.

To discuss effective learning we have to take care of some technicalities. First, the output of a learning algorithm are descriptions of languages instead of languages themselves. Therefore, we extend our notation in Section 2 by replacing languages and language collections by language descriptions and classes of language descriptions. For example, in pattern identification problem, we use PATTERN to denote both the class of all canonical patterns and the language collection it generates; we use $\mathrm{acc}_{\mathrm{PATTERN}}(\mathrm{p})$ to denote the accumulation order of $L(p)$ in the language collection denoted by PATTERN. As another example, we use PATTERN$|S$ to denote both languages consistent with the evidence set S and the patterns that generate them. It should be clear from the context whether we are referring to a language or its description by a pattern that generates the language.

5.2. Mind change optimal identification of one-variable patterns

If a pattern contains exactly one distinct variable (i.e., $\#var(p) = 1$), then it is a **one-variable pattern**. For one-variable patterns, we usually omit the subscript for the variable (e.g., $\mathtt{x01}$ or $\mathtt{0x00x1}$). Following [2], we denote the set of all one-variable patterns by P_1. Angluin described an algorithm that, given a finite set S of strings as input, finds the set of one-variable patterns descriptive of S, and then (arbitrarily) selects one with the maximum length [2, Theorem 6.5]. A one-variable pattern p is **descriptive of a sample** S if $S \subseteq L(p)$ and for every one-variable pattern q such that $S \subseteq L(q)$, the language $L(q)$ is not a proper subset of $L(p)$ [2, p. 48]. To illustrate, the pattern $\mathtt{1x}$ is descriptive of the samples $\{\mathtt{10}\}$ and $\{\mathtt{10}, \mathtt{11}\}$, the pattern $\mathtt{x0}$ is descriptive of the samples $\{\mathtt{10}\}$ and $\{\mathtt{10}, \mathtt{00}\}$, and the pattern $\mathtt{x}$ is descriptive of the sample $\{\mathtt{10}, \mathtt{00}, \mathtt{11}\}$.

We give an example (summarized in Fig. 2) to show that Angluin's algorithm is not an SMC-optimal learner. Let $\mathtt{x}$ be the target pattern and consider a text $T = \langle \mathtt{10}, \mathtt{00}, \mathtt{11}, \mathtt{0}, \ldots \rangle$ for $L(\mathtt{x})$. As mentioned above, we write $P_1|S$ for the set of one-variable patterns consistent with a sample S. Then $P_1|\{\mathtt{10}\} = \{\mathtt{1x}, \mathtt{x0}, \mathtt{x}\}$, $P_1|\{\mathtt{10}, \mathtt{00}\} = \{\mathtt{x0}, \mathtt{x}\}$, $P_1|\{\mathtt{10}, \mathtt{11}\} = \{\mathtt{1x}, \mathtt{x}\}$ and $P_1|\{\mathtt{10}, \mathtt{00}, \mathtt{11}\} = \{\mathtt{x}\}$. The accumulation orders of these languages are determined as follows:

Text T	10	00	11	0	$\cdots$
Stage n	1	2	3	4	$\cdots$
Patterns consistent with $T[n]$	1x, x0, x	x0, x	x	x	$\cdots$
Patterns descriptive of $T[n]$	1x, x0	x0	x	x	$\cdots$
Accumulation order of $T[n]$	1	1	0	0	$\cdots$
Output of Angluin's learner M_A	1x	x0	x	x	$\cdots$
Output of a SMC-optimal learner M	?	x0	x	x	$\cdots$

Fig. 2. An illustration of why Angluin's learning algorithm for one-variable patterns is not strongly mind change optimal.

(1) $\mathrm{acc}_{\mathrm{P_1}}(\mathrm{L}(\mathtt{x})) = 0$ since $L(\mathtt{x})$ is isolated; so $\mathrm{acc}_{\mathrm{P_1}}(\langle \mathtt{10}, \mathtt{00}, \mathtt{11}\rangle) = 0$.
(2) $\mathrm{acc}_{\mathrm{P_1}}(\mathrm{L}(\mathtt{1x})) = 1$ since $P_1|\{\mathtt{10}, \mathtt{11}\} = \{\mathtt{1x}, \mathtt{x}\}$; so $\mathrm{acc}_{\mathrm{P_1}}(\langle \mathtt{10}, \mathtt{11}\rangle) = 1$.
(3) $\mathrm{acc}_{\mathrm{P_1}}(\mathrm{L}(\mathtt{x0})) = 1$ since $P_1|\{\mathtt{10}, \mathtt{00}\} = \{\mathtt{x0}, \mathtt{x}\}$; so $\mathrm{acc}_{\mathrm{P_1}}(\langle \mathtt{10}, \mathtt{00}\rangle) = 1$.

Also, we have $\mathrm{acc}_{\mathrm{P_1}}(\langle \mathtt{10}\rangle) = 1$. Since for $T[1] = \langle \mathtt{10}\rangle$, the one-variable patterns $\mathtt{1x}$ and $\mathtt{x0}$ are both descriptive of $\{\mathtt{10}\}$, an Angluin-style learner M_A conjectures either $\mathtt{1x}$ or $\mathtt{x0}$; suppose $M_A(\langle \mathtt{10}\rangle) = \mathtt{1x}$. Now let c_A be any mind change counter for M_A. Since $\mathtt{1x}$ is consistent with $\langle \mathtt{10}\rangle$, SMC-optimality requires that $c_A(\langle \mathtt{10}\rangle) = \mathrm{acc}_{\mathrm{P_1}}(\langle \mathtt{10}\rangle) = 1$. The next string $\mathtt{00}$ in T refutes $\mathtt{1x}$, so M_A changes its mind to $\mathtt{x0}$ (i.e., $M_A(T[2]) = \mathtt{x0}$), and $c_A(\langle \mathtt{10}, \mathtt{00}\rangle) = 0$. However, M_A changes its mind again to pattern $\mathtt{x}$ on $T[3] = \langle \mathtt{10}, \mathtt{00}, \mathtt{11}\rangle$, so c_A is not a mind change counter for M_A, and M_A is not SMC-optimal. In short, after the string $\mathtt{10}$ is observed, it is possible to identify the target one-variable pattern with one more mind change, but M_A requires two.

The issue with M_A is that M_A changes its mind on sequence $\langle \mathtt{10}, \mathtt{00}\rangle$ even though $\mathrm{acc}_{\mathrm{P_1}}(\langle \mathtt{10}\rangle) = \mathrm{acc}_{\mathrm{P_1}}(\langle \mathtt{10}, \mathtt{00}\rangle) = 1$. Intuitively, a mind change optimal learner has to wait until the data decide between the two patterns $\mathtt{1x}$ and $\mathtt{x0}$. As Proposition 4.1 indicates, we can design an SMC-optimal learner M for P_1 by "procrastinating" with ? until there is a pattern with the highest accumulation order. For example on the text T described above, our SMC-optimal learner M makes the following conjectures: $M(\langle \mathtt{10}\rangle) = ?$, $M(\langle \mathtt{10}, \mathtt{00}\rangle) = \mathtt{x0}$, $M(\langle \mathtt{10}, \mathtt{00}, \mathtt{11}\rangle) = \mathtt{x}$ (see Fig. 2).

The general specification of the SMC-optimal learning algorithm M is as follows. For a terminal $a \in \Sigma$ let $p^a \equiv p[\mathtt{a}/x]$. The proof of [2, Lemma 3.9] shows that if q is a one-variable pattern such that $L(q) \supseteq \{p^a, p^b\}$ for two distinct terminals a, b, then $L(q) \supseteq L(p)$. So if for a pattern p consistent with data σ, the data contain $\{p^a, p^b\}$, then $L(p)$ is a $\subseteq$-minimum for $P_1|\sigma$ and hence has the highest accumulation order for σ. Thus an SMC-optimal learning algorithm M can proceed by waiting until the data feature p^a and p^b for some pattern p. More precisely, define M as follows.

(1) Set $M(\Lambda) := ?$.
(2) Given a sequence σ with $S := \mathrm{content}(\sigma)$, check (*) if there is a one-variable pattern p consistent with σ such that $S \supseteq \{p^a, p^b\}$ for two distinct terminals a, b. If yes, output $M(\sigma) := p$. If not, set $M(\sigma) := ?$.

Since there are at most finitely many patterns consistent with σ, the check (*) is effective.

In fact, (*) and hence M can be implemented so that computing $M(\sigma)$ takes time linear in $|\sigma|$. Outline: Let $m = \min\{|s| : s \in S\}$. Let S^m be the set of strings in S of length m. Define $p_S(i) := a$ if $s(i) = a$ for all $s \in S^m$, and $p_S(i) := \mathtt{x}$ otherwise for $1 \leqslant i \leqslant m$. For example, $p_{\{10,11,111\}} = 1\mathtt{x}$ and $p_{\{10,01\}} = \mathtt{x}$. Then check for all $s \in S$ if $s \in L(p_S)$. For a one-variable pattern, this can be done in linear time because $|\theta(\mathtt{x})|$, the length of $\theta(\mathtt{x})$, must be $\frac{|s|-term(p_S)}{|p_S|-term(p_S)}$ where $term(p_S)$ is the number of terminals in p_S. For example, if $s = 111$ and $p_S = 1\mathtt{x}$, then $|\theta(\mathtt{x})|$ must be 2. If p_S is consistent with S, then there are distinct $a, b \in \Sigma$ such that $\{p^a, p^b\} \subseteq S$. Otherwise no pattern p of length m is consistent with S and hence (*) fails.

It is worth noting that sometimes the mind change efficient learner M_{P_1} may take longer to converge than the Angluin-style learner M_A. For example, let T be a text for the pattern $1\mathtt{x}$ such that $T(0) = 10$ and $T(1) = 11$; then we can verify that the Angluin-style learner M_A in Fig. 2 converges at time 0, but a mind change efficient learner does not converge until time 1. In general, an Angluin-style learner will converge to the correct one-variable pattern at least as soon as a SMC-optimal learner and strictly sooner on some texts. Thus, the Angluin-style learner dominates the SMC-learner with respect to convergence time in the sense of [27] and [17].

5.3. Mind change optimal identification of fixed-length patterns

Following [31], for each positive integer n, we write T_n to denote the set of canonical patterns of length n. We apply the concept of accumulation order to design a mind change efficient algorithm that identifies T_n for a fixed n. The first step is to find an easily computable, closed-form expression for the accumulation order of a pattern in T_n.

Lemma 5.1. *Fix a positive integer $n > 0$, and let p be a pattern in T_n. Then* $\mathrm{acc}_{\mathrm{T_n}}(\mathrm{p}) = \mathrm{n} - \#\mathrm{var}(\mathrm{p})$, *where $\#var(p)$ is the number of distinct variables in p.*

Proof. We prove the claim by downward induction.

Base case: $\#var(p) = n$. Then p is the most general pattern $\mathtt{x}_1\mathtt{x}_2\cdots\mathtt{x}_n$; thus $\mathrm{acc}_{\mathrm{T_n}}(\mathrm{p}) = 0$.

Inductive step: Assume $\mathrm{acc}_{\mathrm{T_n}}(\mathrm{q}) = \mathrm{n} - \#\mathrm{var}(\mathrm{q})$ for all q with $\#var(q) > k$. Consider a pattern p with $\#var(p) = k$. Let $r \in T_n$ be another pattern of length n. If $\#var(r) < k$, then $|S(r)| < |S(p)|$ so $S(p) \not\subseteq S(r)$. Angluin shows that $S(p) = S(r)$ implies $L(p) = L(r)$ [2, Lm. 3.2]. So if $\#var(r) = k$ and $L(p) \neq L(r)$, then $S(p) \neq S(r)$ and so $S(p) \not\subseteq S(r)$ since $|S(p)| = |S(r)|$. So in either case, $S(p) \not\subseteq S(r)$. As there are only finitely many patterns of length n, this implies that there exists a finite subset $S \subseteq L(p)$ such that $L(r) \neq L(p)$ implies that $\#var(r) > k$ for every pattern $r \in T_n|S$. By the induction assumption, it follows that (1) $\mathrm{acc}_{\mathrm{T_n}}(\mathrm{p}) \leqslant \mathrm{n} - \mathrm{k}$.

Second, since $\#var(p) < n$, it is easy to see that there exists a pattern r such that $\#var(r) = \#var(p) + 1$ and $q \succeq p$; thus $L(p) \subseteq L(q)$. This implies that (2) $\mathrm{acc}_{\mathrm{T_n}}(\mathrm{p}) \geqslant \mathrm{n} - (\mathrm{k} - 1) + 1 = \mathrm{n} - \mathrm{k}$. Combining the above two inequalities (1) and (2), we have $\mathrm{acc}_{\mathrm{T_n}}(\mathrm{p}) = \mathrm{n} - \mathrm{k} = \mathrm{n} - \#\mathrm{var}(\mathrm{p})$. □

To illustrate, the lemma implies that $\mathrm{acc}_{\mathrm{T_n}}(\mathtt{x}_1\mathtt{x}_2\cdots\mathtt{x}_n) = 0$, $\mathrm{acc}_{\mathrm{T_n}}(\mathtt{x}_1 0\mathtt{x}_2\cdots\mathtt{x}_{n-1}) = 1$, and $\mathrm{acc}_{\mathrm{T_n}}(\mathtt{x}_1\mathtt{x}_1\cdots\mathtt{x}_1) = \mathrm{n} - 1$.

Lemma 5.1 allows us to design a strongly mind change optimal learner as follows. First, we observe that every data sequence σ is topped for the language collection T_n. This is because T_n is finite. For a finite set of ordinals $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$, its supremum is its maximum. Thus Condition 2 of Proposition 4.1 holds vacuously. Condition 1 requires a strongly mind change optimal learner to

output ? or the pattern that uniquely tops the given data sequence σ. For a given data sequence σ, we can enumerate the finitely many patterns $T_n|\sigma$ of length n that are consistent with the strings in σ. Then we simply check if any pattern p in $T_n|\sigma$ uniquely maximizes $n - \#var(p)$ or equivalently minimizes $\#var(p)$.

In principle, closed form expressions for the accumulation order of a pattern p in the one-variable pattern space P_1 and in the general pattern space PATTERN, such as Lemma 5.1 provides for T_n, would yield mind change optimal learners for these language collections. Finding closed form expressions for $\mathrm{acc}_{\mathrm{P_1}}$ and $\mathrm{acc}_{\mathrm{PATTERN}}$ are currently open problems [26].

6. Accumulation order and structural complexity

Our final section relates accumulation order to other well-known learning-theoretic concepts that describe the structure of a learning problem.

6.1. Thickness and inclusion depth

It follows from Clause 2 of Lemma 3.1 that the accumulation order of a language L in a language collection $\mathcal{L}$ is at least as great as the length of a chain of supersets of L. We refer to this length as the inclusion depth of $\mathcal{L}$, as formalized the following definition.

Definition 6.1. Let $\mathcal{L}$ be a language collection and L be a language in $\mathcal{L}$. The **inclusion depth** of L in $\mathcal{L}$ is the size n of the largest index set $\{L_i\}_{1\leqslant i\leqslant n}$ of distinct languages in $\mathcal{L}$, such that $L \subset L_1 \subset \ldots \subset L_i \subset \ldots \subset L_n$. The **inclusion depth** of $\mathcal{L}$ is the maximum of the inclusion depths of languages in $\mathcal{L}$ (cf. [26]).

For example, in COINIT, the inclusion depth of language $L_n = \{i \in \mathbb{N} : i \geqslant n\}$ is n. The inclusion depth of COINIT is ω. For many language collections, the inclusion depth of a language L is not only a lower bound on its accumulation order but characterizes it exactly. As we will show, examples include COINIT, LINEAR_n, P_1, T_n, and PATTERN. The following proposition shows that a fairly simple property due to Angluin [3, Condition 3] is a sufficient condition for the accumulation order of a language to be equal to its inclusion depth. Following [41], we say that a class of languages $\mathcal{L}$ has **finite thickness** if $\mathcal{L}|\{s\}$ is finite for every string $s \in \bigcup \mathcal{L}$. Note that if the language collection $\mathcal{L}$ has finite thickness, then every language in $\mathcal{L}$ has finite inclusion depth, so the inclusion depth of $\mathcal{L}$ is at most ω.

In language collections of finite thickness, the inclusion depth of a language is exactly its accumulation order.

Proposition 6.1. *Let $\mathcal{L}$ be a language collection with finite thickness and L be a language in $\mathcal{L}$.*

(1) *There is a finite subset $S \subseteq L$ such that L is a $\subseteq$-minimum in $\mathcal{L}|S$.*
(2) *The inclusion depth of L is* $\mathrm{acc}_{\mathcal{L}}(\mathrm{L})$.

Proof. For clause 1, let s be a string in L; then $\mathcal{L}|\{s\}$ is finite since $\mathcal{L}$ has finite thickness. For every language $L' \in \mathcal{L}|\{s\}$ such that $L' \not\supseteq L$, the set $L \setminus L'$ is nonempty. For each L', choose a string $s_{L'}$ from $L \setminus L'$, and let $S := \{s\} \cup \{s_{L'} : L' \in \mathcal{L}|\{s\} \setminus \{L\}\}$. Then $\mathcal{L}|S$ contains only languages that include L.

We prove clause 2 by induction.

Base case: Let L be a language with the inclusion depth 0, which means that there is no language that properly includes L. Then there exists a finite set $S \subseteq L$ such that $\mathcal{L}|S = \{L\}$. Therefore, $\mathrm{acc}_{\mathcal{L}}(\mathrm{L}) = 0$ by the definition of accumulation order.

Inductive step: Assume for every language with inclusion depth less than k that its accumulation order equals its inclusion depth. Consider the case that L has inclusion depth k. From the induction assumption, we know that there exists a language L' such that $L \subset L'$ and $\mathrm{acc}_{\mathcal{L}}(\mathrm{L}') = \mathrm{k} - 1$. Therefore, (1) $\mathrm{acc}_{\mathcal{L}}(\mathrm{L}) \geqslant \mathrm{k}$ by Clause 2 of Lemma 3.1. On the other hand, since $\mathcal{L}$ has finite thickness, there exists a subset $S \subseteq L$ such that $\mathcal{L}|S$ contains only languages that include L. It is clear that for every language $L' \in \mathcal{L}|S$, if $L' \neq L$ then $L' \supset L$; this implies that L' has inclusion depth less than k for every language $L' \in \mathcal{L}|S - L$, otherwise L would have inclusion depth greater than k. Therefore, $\mathrm{acc}((\mathcal{L}|\mathrm{S}) \setminus \{\mathrm{L}\}) = \sup\left(\mathrm{acc}_{\mathcal{L}}\{\mathrm{L}' \in \mathcal{L}|\mathrm{S} \setminus \{\mathrm{L}\}\}\right) < \mathrm{k}$; thus (2) $\mathrm{acc}_{\mathcal{L}}(\mathrm{L}) \leqslant \mathrm{k}$. Combining the two inequalities (1) and (2), we have $\mathrm{acc}_{\mathcal{L}}(\mathrm{L}) = \mathrm{k}$, which complete the inductive step. □

As it is easy to verify that each of the language collections COINIT, LINEAR_n, P_1, T_n, and PATTERN has finite thickness, the proposition implies that the accumulation order of each language L in these collections is the inclusion depth of L, or the maximum length of a chain of supersets of L. Clause 1 of the proposition establishes that in languages with finite thickness, the general strategy of conjecturing $\subseteq$-minima is sufficient for constructing SMC-optimal learners.

6.2. Elasticity

We show that the concept of elasticity provides a sufficient condition for a language collection $\mathcal{L}$ to have a bounded accumulation order, which by Theorem 3.1 implies that $\mathcal{L}$ is identifiable with a bounded number of mind changes.

A class of languages $\mathcal{L}$ has **infinite elasticity** if there exist an infinite sequence of strings $(s_i)_{i\in\mathbb{N}}$ and an infinite sequence of languages $(L_i)_{i\in\mathbb{N}}$, where $L_i \in \mathcal{L}$, such that for each $i \in \mathbb{N}$, $\{s_0, \ldots, s_i\} \subseteq L_i$ but $s_{i+1} \notin L_i$. A class of languages has **finite elasticity** iff it does not have infinite elasticity [44], [30, Def. 7]. For example the language collection LINEAR_n has finite elasticity because if vector $\vec{v}_{i+1}$ is not in a linear subspace L_i, then $\vec{v}_{i+1}$ is independent of any subset $\{\vec{v}_0, \vec{v}_1, \ldots, \vec{v}_i\}$. It is not hard to see that finite thickness implies finite elasticity [44], so T_n and P_1 have finite elasticity.

We use $D(\mathcal{L})$ to denote all finite subsets of languages in $\mathcal{L}$. A subset $\mathcal{P}$ of a topological space is **perfect** if $\mathcal{P}$ has no isolated points [23, p. 78].

Lemma 6.1. *Let $\mathcal{P}$ be a perfect nonempty set of languages. Then $\mathcal{P}|d$ is also nonempty and perfect for every finite subset $d \in D(\mathcal{P})$.*

To illustrate, the language collection FIN which comprises all finite subsets of $\mathbb{N}$ is perfect and nonempty. Since no finite subset d entails a single language in FIN (i.e., $\mathrm{card}(\mathrm{FIN}|d) > 1$), we have that $\mathrm{FIN}|d$ is nonempty and perfect.

The next proposition gives a topological condition sufficient to establish that a language collection $\mathcal{L}$ has infinite elasticity, namely that $\mathcal{L}$ contain a subset that is perfect in the language topology for $\mathcal{L}$. If $\mathcal{L}$ has a perfect subset, the derivation procedure from Section 3.1 terminates with a nonempty perfect kernel, and $\mathcal{L}$ has no bounded accumulation order, which by Theorem 3.1 is equivalent to the statement that $\mathcal{L}$ is not identifiable with an ordinal mind change bound. Contrapositively, if $\mathcal{L}$ has finite elasticity, then $\mathcal{L}$ is identifiable with an ordinal mind change bound.

Proposition 6.2. *Let $\mathcal{L}$ be a collection of languages.*

(1) *If $\mathcal{L}$ contains a nonempty perfect subset $\mathcal{P} \subseteq \mathcal{L}$, then $\mathcal{L}$ has infinite elasticity.*
(2) *If $\mathcal{L}$ has finite elasticity, then $\mathcal{L}$ has a bounded accumulation order and hence $\mathcal{L}$ is identifiable with a bounded number of mind changes.*

Proof. Part 1: If $\mathcal{P} \neq \emptyset$ is perfect, then $\mathcal{P}$ is infinite and so there are infinitely many languages $L \in \mathcal{P}$ such that $L \neq \bigcup \mathcal{P}$. Choose a nonempty language $L_0 \neq \bigcup \mathcal{P}$ and strings $s_0 \in L_0$ and $s_1 \in \bigcup \mathcal{P} - L_0$. Let $\mathcal{P}_1 := \mathcal{P}|\{s_0, s_1\}$. Then by Lemma 6.1, $\mathcal{P}_1$ is a nonempty perfect set. So there is nonempty $L_1 \in \mathcal{P}_1$ such that $L_1 \neq \bigcup \mathcal{P}_1$, and we may choose a string $s_2 \in \bigcup \mathcal{P}_1 - L_1$. Continuing this process indefinitely, we obtain two sequences $(L_i)_{i\in\mathbb{N}}$ and $(s_i)_{i\in\mathbb{N}}$ such that for each $i \in \mathbb{N}$, $\{s_0, \ldots, s_i\} \subseteq L_i$ but $s_{i+1} \notin L_i$. In other words, $\mathcal{L}$ has infinite elasticity.

Part 1: Suppose that $\mathcal{L}$ has finite elasticity. Then by the contrapositive of Clause 1, the only perfect subset of $\mathcal{L}$ is the empty set. Since the derivation procedure from Definition 3.1 terminates with a perfect subset of $\mathcal{L}$, it thus terminates with the empty set, so $\mathcal{L}$ is scattered and has bounded accumulation order by Corollary 3.1. □

The proposition implies that LINEAR_n and all sub-collections of PATTERN are identifiable with a bounded number of mind changes.

If a language has infinite elasticity, then it also has infinite thickness. It is known that, for indexed language families, finite elasticity is a sufficient condition for effective learnability [44,30]. A sequence of nonempty languages $\{L_i\}$ constitutes an indexed family just in case there exists a computable function f such that for each $i \in N$ and for each $x \in N$, we have $f(i,x) = 1$ if $x \in L_i$ and $f(i,x) = 0$ otherwise [3, Section 2], [12, Ex. 4.7]. Fig. 3 illustrates the relationship among these structural concepts.

6.3. Intrinsic complexity

Next we consider the relationship between weak and strong reducibility, intrinsic complexity [10,13], and accumulation order. Our basic result is that if language collection $\mathcal{L}_1$ is reducible to $\mathcal{L}_2$, then $\text{acc}(\mathcal{L}_2) \geqslant \text{acc}(\mathcal{L}_1)$. In this sense reducibility agrees with accumulation order—and hence mind change complexity—as a comparison of the complexity of different learning problems.

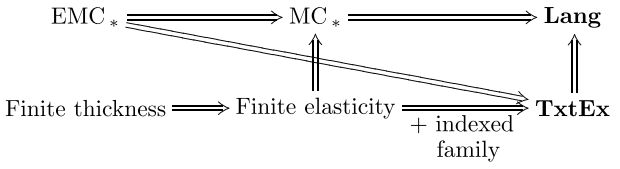

Fig. 3. Relations between various computable and noncomputable identifiability concepts. *EMC*$_*$ denotes language collections identifiable by a computable learner with a bounded number of mind changes. MC_* denotes language collections with bounded accumulation orders, or equivalently, identifiable by a noncomputable learner with a bounded number of mind changes. Following [12], we use **Lang** to denote all language collections identifiable by noncomputable learners and use **TxtEx** to denote all language collections identifiable by computable learners. The notation $\Rightarrow_{+\text{ indexed family}}$ indicates that the implication holds only for indexed language collections.

Definition 6.2 (*[15,13,10]*).

(1) An **enumeration operator** is a computable function that maps SEQ into SEQ.
(2) An infinite sequence G is **admissible** for a text T if G converges to an index (or grammar) of the language $L = \text{content(T)}$.
(3) Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two classes of languages. Then $\mathcal{L}_1$ is **weakly reducible to** $\mathcal{L}_2$, denoted by $\mathcal{L}_1 \leqslant_{weak} \mathcal{L}_2$, if there exist two enumeration operators Θ and Ξ such that for every text T_1 for $\mathcal{L}_1$,
   (a) $\Theta(T_1) = \bigcup_n \Theta(T_1[n])$ is a text for $\mathcal{L}_2$.
   (b) for every admissible sequence G for $\Theta(T_1)$, the sequence $\Xi(G) = \bigcup_n \Xi(G[n])$ is admissible for T_1.

   We say that operators Θ and Ξ **witness** $\mathcal{L}_1 \leqslant_{weak} \mathcal{L}_2$.
(4) Language collection $\mathcal{L}_1$ is **strongly reducible to** $\mathcal{L}_2$, denoted by $\mathcal{L}_1 \leqslant_{strong} \mathcal{L}_2$, if there exists Θ and Ξ such that
   (a) Θ and Ξ witness $\mathcal{L}_1 \leqslant_{weak} \mathcal{L}_2$, and
   (b) for every language $L_1 \in \mathcal{L}_1$, there exists a language $L_2 \in \mathcal{L}_2$ such that $L_2 = \text{content}(\Theta(\text{T}))$ for every text T for L_1.

The following proposition relates accumulation order to reducibility.

Proposition 6.3. *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two language collections such that $\mathcal{L}_1 \leqslant_{weak} \mathcal{L}_2$ is witnessed by operators Θ and Ξ.*

(1) *Let L and L' be two distinct languages in $\mathcal{L}_1$, and let T and T' be texts for L and L', respectively. Then* $\text{content}(\Theta(\text{T})) \neq \text{content}(\Theta(\text{T}'))$. *(Thus, texts from distinct languages are mapped to texts from distinct languages.)*
(2) *Let T be a text for some $L_1 \in \mathcal{L}_1$, and let $L_2 = \text{content}(\Theta(\text{T}))$. Then* $\text{acc}_{\mathcal{L}_2}(\text{L}_2) \geqslant \text{acc}_{\mathcal{L}_1}(\text{L}_1)$.
(3) *If $\mathcal{L}_1 \leqslant_{weak} \mathcal{L}_2$, then* $\text{acc}(\mathcal{L}_1) \leqslant \text{acc}(\mathcal{L}_2)$. *Therefore if $\mathcal{L}_2$ is identifiable with mind change bound α, so is $\mathcal{L}_1$.*

Proof. Clause 1: For contradiction, assume $\text{content}(\Theta(\text{T})) = \text{content}(\Theta(\text{T}')) = \text{L}_2 \in \mathcal{L}_2$. If G is an admissible sequence for $\Theta(T)$, then G is also an admissible sequence for $\Theta(T')$. Therefore $\Xi(G)$ is admissible for both T and T', which is impossible.

Clause 2: The proof is by transfinite induction on $\text{acc}_{\mathcal{L}_2}(\text{L}_2)$. Assume the claim hold for all cases where $\text{acc}_{\mathcal{L}_2}(\text{L}_2) = \beta < \alpha$, and suppose that $\text{acc}_{\mathcal{L}_2}(\text{L}_2) = \alpha$.

For contradiction, assume that $\text{acc}_{\mathcal{L}_1}(\text{L}_1) = \gamma > \alpha$. Since $\Theta(T)$ is a text for L_2, by Lemma 3.1, there exists a time n such that L_2 uniquely has the highest accumulation order α in $\mathcal{L}_2|\Theta(T)[n]$. Let m be a time such that (a) $\Theta(T[m]) \supseteq \Theta(T)[n]$. Since T is a text for L_1 and $\text{acc}_{\mathcal{L}_1}(\text{L}) > \alpha$, there is a language $L_1' \in \mathcal{L}_1|T[m]$ such that $\text{acc}_{\mathcal{L}_1}(\text{L}_1') = \alpha$. Let T' be a text for L_1' that extends $T[m]$; then by Clause 6.3 we have that $\text{content}(\Theta(\text{T}')) \neq \text{L}_2$. Let us write L_2' for $\text{content}(\Theta(\text{T}'))$. Clearly $\text{content}(\Theta(\text{T}'[\text{m}])) \subseteq \text{content}(\Theta(\text{T}')) = \text{L}_2'$, so $L_2' \in \mathcal{L}_2|\Theta(T'[m])$. Since $T[m] = T'[m]$ we have (b) $L_2' \in \mathcal{L}_2|\Theta(T[m])$. Combining (a) and (b) we have (c) $L_2' \in \mathcal{L}_2|\Theta(T)[n]$.

Since L_2 is the only language in $\mathcal{L}_2|\Theta(T)[n]$ with the accumulation order α, the language L'_2 must has an accumulation order $\beta < \alpha$ in $\mathcal{L}_2$. Therefore, $\mathrm{acc}_{\mathcal{L}_1}(\mathrm{L}'_1) = \alpha > \mathrm{acc}_{\mathcal{L}_2}(\mathrm{L}'_2) = \beta$, which contradicts the induction hypothesis and establishes the inductive step.

Clause 3: Immediate consequence of Clause 2. □

The above proposition gives us a necessary condition for reducibility, which we illustrate in the following examples. As in [13], SINGLE denotes the class of all singleton languages. It is easy to see that acc(COINIT) = ω but acc(SINGLE) = 0, therefore COINIT $\not\leqslant_{weak}$ SINGLE, as shown in [13].

If $\mathcal{L}_1$ is not scattered (i.e., has no mind change bound) and $\mathcal{L}_2$ is scattered (i.e., has a mind change bound), then Proposition 6.3 implies that $\mathcal{L}_1$ is not weakly reducible to $\mathcal{L}_2$. Since the class of all finite languages FIN is not scattered (cf. Section 3.1), it follows that FIN $\not\leqslant_{weak}$ COINIT, as established by [13].

If Θ and Ξ witness $\mathcal{L}_1 \leqslant_{strong} \mathcal{L}_2$, then Θ induces a function f_Θ that maps $\mathcal{L}_1$ into $\mathcal{L}_2$ as follows: for a language $L \in \mathcal{L}_1$, choose any text T for L, and assign $f(L) = \mathrm{content}(\Theta(\mathrm{T}))$. The definition of strong reducibility guarantees that f_Θ is well-defined. We show that f_θ is a continuous one-one function in our topology. A function $f : X \to Y$ is **continuous** if for every point $x \in X$ and every neighborhood V of $f(x)$ in Y, there exists a neighborhood U of x in X, such that $f(U) \subseteq V$.[7] For two language collections $\mathcal{L}_1$ and $\mathcal{L}_2$, this means that $f : \mathcal{L}_1 \to \mathcal{L}_2$ is continuous if for every language $L_1 \in \mathcal{L}_1$ and every finite subset $D_2 \subseteq f(L_1)$, there is a finite subset $D_1 \subseteq L_1$ such that $\{f(L) : L \in \mathcal{L}_1|D_1\} \subseteq \mathcal{L}_2|D_2$.

Lemma 6.2. *Suppose* Θ *and* Ξ *witness* $\mathcal{L}_1 \leqslant_{strong} \mathcal{L}_2$. *Then* $f_\Theta : \mathcal{L}_1 \to \mathcal{L}_2$ *defined above is a continuous one-one function.*

The proof is left to the reader.

Lemma 6.2 connects strong reducibility with many basic results in point-set topology. As an illustration, we apply standard theorems in topology to immediately derive that strong reducibility respects accumulation order without the need for the construction of Proposition 6.3.

Proposition 6.4. *Let* $f : X \mapsto Y$ *be a continuous one-one function, and let* $A \subseteq X$ *and* $x \in X$.

(1) *If* $x \in A^{(1)}$, *then* $f(x) \in f(A)^{(1)}$ (*i.e.*, $f(A^{(1)}) \subseteq [f(A)]^{(1)}$).
(2) *If* acc(Y) *is defined, then* acc(X) *is also defined and moreover* acc(X) $\leqslant$ acc(Y).

Proof. Clause 1 is Theorem 2.3 of [7]. Clause 2 follows easily by transfinite induction. □

Therefore we can establish the following result from standard topological results.

Corollary 6.1. *Suppose* Θ *and* Ξ *witness* $\mathcal{L}_1 \leqslant_{strong} \mathcal{L}_2$. *Then* $\mathrm{acc}(\mathcal{L}_1) \leqslant \mathrm{acc}(\mathcal{L}_2)$.

Thus if $f_\Theta : \mathcal{L}_1 \to \mathcal{L}_2$ is onto and $(f_\Theta)^{-1} : \mathcal{L}_2 \to \mathcal{L}_1$ is continuous, then $\mathrm{acc}(\mathcal{L}_1) = \mathrm{acc}(\mathcal{L}_2)$; in topological terminology, homeomorphic language collections have the same accumulation order.

[7] This definition is equivalent to the condition that $f^{-1}(V)$ is open in X for every open set V of Y.

7. Summary and future work

The topic of this paper was learning with bounded mind changes. We applied the classic topological concept of accumulation order to characterize the mind change complexity of a learning problem: A language collection $\mathcal{L}$ is identifiable by a learner (not necessarily computable) with α mind changes iff the accumulation order of $\mathcal{L}$ is at most α. We studied the properties of strongly mind change optimal learners: roughly, a learner Ψ is strongly mind change optimal if Ψ realizes the best possible mind change bound not only in the entire learning problem, but also in subproblems that arise after observing some data. The characteristic property of SMC-optimal learners is that they output languages with maximal accumulation order. Thus, analyzing the accumulation order of a learning problem is a powerful guide to constructing mind change efficient learners. We illustrated these results in several learning problems such as identifying a linear subspace and one-variable and fixed-length patterns. For learning linear subspaces, the natural method of conjecturing the least subspace containing the data is the only mind change optimal learner that does not "procrastinate" (i.e., never outputs ? or an inconsistent conjecture). This is exactly the inference procedure that the particle physics community has followed to arrive at the set of conservation laws found in the current standard model of particle physics. Angluin's algorithm for learning a one-variable pattern is not SMC-optimal; we described a different SMC-optimal algorithm for this problem that has linear update time.

An interesting open issue in the general theory of SMC-optimal learning is the relationship between mind change optimality and time efficiency. As the example of one-variable patterns shows, there can be a trade-off between time efficiency and producing consistent conjectures, on the one hand, and the procrastination that minimizing mind changes may require on the other (see Section 5). We would like to characterize the learning problems for which this tension arises, and how great the trade-off can be. For example, if a language collection $\mathcal{L}$ is closed under intersection, then conjecturing $\cap(\mathcal{L}|\sigma)$ for every data sequence σ yields an SMC-optimal learner that never procrastinates (the so-called "closure algorithm" [5]). The language collection LINEAR and the learner Ψ_{LIN} are an instance of an intersection-closed language class and the corresponding closure algorithm. Are there are other general sufficient or necessary conditions for a procrastination-free SMC-optimal learner?

As we have seen, mind change optimality imposes strong constraints on learners. This means that we can apply our theory to design optimal learning algorithms for problems of interest. Such an analysis can validate existing inference procedures, as in the case of learning conservation laws, or lead to the development of new ones, as with one-variable patterns. Other potential applications include the following. The next challenge for pattern languages is to find an SMC-optimal algorithm for learning a general pattern with arbitrarily many variables. An important step towards that goal would be to determine the accumulation order of a pattern language $L(p)$ in the space of pattern languages [26]. Another application is the design of SMC-optimal learners for logic programs. For example, Jain and Sharma have examined classes of logic programs that can be learned with bounded mind changes using explorer trees [14]. Do explorer trees lead to mind change optimal learning algorithms? One approach to learning causal graphs or Bayes nets is based on independence relations extracted from the data, where the graph is viewed as a compact representation of these independence facts [35,36]. What are mind change optimal algorithms that identify a correct graph in the limit from independence data?

In sum, strong mind change optimality guides the construction of learning algorithms by imposing strong and natural constraints; and the analytical tools we established for solving these constraints reveal significant aspects of the fine structure of learning problems.

Acknowledgments

This research was supported by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada. We thank the anonymous referee for valuable comments and suggestions.

References

[1] A. Ambainis, S. Jain, A. Sharma, Ordinal mind change complexity of language identification, Theor. Comput. Sci. 220 (2) (1999) 323–343.
[2] D. Angluin, Finding patterns common to a set of strings, J. Comput. Syst. Sci. 21 (1) (1980) 46–62.
[3] D. Angluin, Inductive inference of formal languages from positive data, Inf. Control 45 (2) (1980) 117–135.
[4] K. Apsitis, Derived sets and inductive inference, in: S. Arikawa, K.P. Jantke (Eds.), Proceedings of ALT 1994, Springer, Berlin, Heidelberg, 1994, pp. 26–39.
[5] P. Auer, R. Ortner, A new pac bound for intersection-closed concept classes., in: COLT, 2004, pp. 408–414.
[6] G. Baliga, J. Case, S. Jain, The synthesis of language learners, Inf. Comput. 152 (1999) 16–43.
[7] J.D. Baum, Elements of Point Set Topology, rep. Dover, 1991, 1964.
[8] G. Cantor, Grundlagen einer allgemeinen Mannigfaltigkeitslehre, in: W. Ewald (Ed.), From Kant to Hilbert, vol. 2, Oxford Science Publications, 1996, pp. 878–920.
[9] R. Freivalds, C.H. Smith, On the role of procrastination in machine learning, Inf. Comput. 107 (2) (1993) 237–271.
[10] R. Freivalds, E. Kinber, C.H. Smith, On the intrinsic complexity of learning, Inf. Comput. 123 (1) (1995) 64–71.
[11] E.M. Gold, Language identification in the limit, Inf. Control 10 (5) (1967) 447–474.
[12] S. Jain, D. Osherson, J.S. Royer, A. Sharma, Systems That Learn, second ed., MIT Press, Cambridge, MA, 1999.
[13] S. Jain, A. Sharma, The intrinsic complexity of language identification, J. Comput. Syst. Sci. 52 (3) (1996) 393–402.
[14] S. Jain, A. Sharma, Mind change complexity of learning logic programs, TCS 284 (1) (2002) 143–160.
[15] S. Jain, E.B. Kinber, R. Wiehagen, Language learning from texts: degrees of intrinsic complexity and their characterizations., J. Comput. Syst. Sci. 63 (3) (2001) 305–354.
[16] A.J. Jayanthan, Derived length for arbitrary topological spaces, Int. J. Math. Math. Sci. 15 (2) (1992) 273–277.
[17] K. Kelly, The Logic of Reliable Inquiry, Oxford University Press, 1996.
[18] K. Kelly, O. Schulte, The computable testability of theories with uncomputable predictions, Erkenntnis 43 (1995) 29–66.
[19] K. Kelly, Efficient convergence implies Ockham's Razor, in: Proceedings of the 2002 International Workshop on Computation Models of Scientific Reasoning and Applications, 2002, pp. 24–27.
[20] K. Kelly, A close shave with realism: Ockham's razor derived from efficient convergence, 2003 (completed manuscript) Available http://www.andrew.cmu.edu/user/kk3n/kelly/ockham.pdf.
[21] K. Kelly, Justification as truth-finding efficiency: how ockham's razor works, Minds Mach. 14 (4) (2004) 485–505.
[22] S. Kocabas, Conflict resolution as discovery in particle physics, Mach. Learn. 6 (1991) 277–309.
[23] K. Kuratowski, Topology, vol. 1, Academic Press, 1966 (translated by J. Jaworowski).
[24] S. Lange, T. Zeugmann, Language learning with a bounded number of mind changes, in: Symposium on Theoretical Aspects of Computer Science, 1993, pp. 682–691.
[25] W. Luo, O. Schulte, Mind change efficient learning., in: P. Auer, R. Meir (Eds.), Learning Theory, 18th Annual Conference on Learning Theory, COLT 2005, Bertinoro, Italy, June 27–30, 2005, Proceedings, Lecture Notes in Computer Science, vol. 3559, Springer, 2005, pp. 398–412.

[26] W. Luo, Compute inclusion depth of a pattern., in: P. Auer, R. Meir (Eds.), Learning Theory, 18th Annual Conference on Learning Theory, COLT 2005, Bertinoro, Italy, June 27–30, 2005, Proceedings, Lecture Notes in Computer Science, vol. 3559, Springer, 2005, pp. 689–690.
[27] E. Martin, D.N. Osherson, Elements of Scientific Inquiry, MIT Press, Cambridge, MA, 1998.
[28] E. Martin, A. Sharma, F. Stephan, Learning, logic, and topology in a common framework, in: Proceedings of the 13th International Conference on Algorithmic Learning Theory, Springer-Verlag, 2002, pp. 248–262.
[29] E. Martin, A. Sharma, On a syntactic characterization of classification with a mind change bound, in: COLT 2005, Lecture Notes in Computer Science, vol. 3559, Springer, 2005, pp. 413–428.
[30] T. Motoki, T. Shinohara, K. Wright, The correct definition of finite elasticity: corrigendum to identification of unions, in: Proceedings of COLT 1991, Morgan Kaufmann Publishers, 1991, p. 375.
[31] Y. Mukouchi, Characterization of pattern languages, in: Proc. 2nd workshop on algorithmic learning theory (ALT'91), 1991, pp. 93–104.
[32] Y. Mukouchi, Inductive inference with bounded mind changes, in: S. Doshita, K. Furukawa, K.P. Jantke, T. Nishida (Eds.), Proceedings of ALT 1992, Springer, Berlin, Heidelberg, 1993, pp. 125–134.
[33] P. Odifreddi, Classical Recursion Theory, North-Holland, New York, 1999.
[34] D.N. Osherson, M. Stob, S. Weinstein, Systems That Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists, MIT Press, Cambridge, MA, 1986.
[35] J. Pearl, Causality: Models, Reasoning, and Inference, Cambridge University Press, 2000.
[36] P. Spirtes, C. Glymour, R. Scheines, Causation, Prediction, and Search, MIT Press, Cambridge, MA, 2000.
[37] G. Plotkin, A note on inductive generalization, in: Machine Intelligence, vol. 5, Edinburgh University Press, 1970, pp. 153–163.
[38] G. Plotkin, A further note on inductive generalization, in: Machine Intelligence, vol. 6, Edinburgh University Press, 1971, pp. 101–124.
[39] O. Schulte, Automated discovery of conservation principles and new particles in particle physics, Mach. Learn. (2005) (submitted).
[40] O. Schulte, An algorithmic proof that the family conservation laws are optimal for the current reaction data, arXiv preprint archive. Available from: <http://arxiv.org/abs/hep-ph/0602011>.
[41] T. Shinohara, Inductive inference of monotonic formal systems from positive data, New Gen. Comput. 8 (4) (1991) 371–384.
[42] R. Valdés-Pérez, Algebraic reasoning about reactions: discovery of conserved properties in particle physics, Mach. Learn. 17 (1994) 47–67.
[43] R. Valdés-Pérez, On the justification of multiple selection rules of conservation in particle physics phenomenology, Comput. Phys. Commun. 94 (1996) 25–30.
[44] K. Wright, Identification of unions of languages drawn from an identifiable class, in: Proceedings of the second annual workshop on Computational learning theory, Morgan Kaufmann Publisher, 1989, pp. 328–333.