Algorithmic Learning Theory

JENNIFER CHUBB

UNIVERSITY OF SAN FRANCISCO

Computably enumerable sets

Definition. A set is computably enumerable if it is the domain of a PCF. W_e is the domain of φ_e.

Theorem. A non-empty set is computably enumerable if and only if it is the range of a computable function.

Computable Sets

Definition. A set is computable if its characteristic function is computable.

Theorem. Let $A \subseteq \mathbb{N}$. Then A is computable if and only if both A and $\overline{A}$ are computably enumerable.

Names and notation

Notations.

- φ_e is the function computed by TM with index e.
- $\varphi_{e,s}(n)$ is $\varphi_e(n)$ if that computation halts in fewer than s steps, and is -1 otherwise.
- W_e is the domain of φ_e, also called *the eth c.e. set*.
- $\langle x, y\rangle$ is a number that encodes the ordered pair (x, y).

Names and notation

Notations.

- RE is the set of all computably enumerable sets.
- RE_{REC} is the set of all computable sets.
- $\mathcal{F}$ is the set of all functions (partial or total, partially computable or not) on $\mathbb{N}$.
- $\mathcal{F}^{REC}$ is the set of all computable functions on $\mathbb{N}$.

Questions

Question 1. Are there sets that are not computably enumerable?

- ✓ A cardinality argument or diagonalization proves that there are.

Question 2. Are there sets that **are** computably enumerable but **are not** computable?

Is $RE_{REC} \subsetneq RE$?

The Halting Set

Definition. $\mathbb{K} = \{e \in \mathbb{N} \mid \varphi_e(e)\downarrow\}$.

The Halting Set is computably enumerable.

- It is the domain of a PCF.
- It is the range of a CF (such a function enumerates it).

Theorem. The Halting Set is NOT computable.

Corollary. The set $\mathbb{K}_0 = \{\langle e, n\rangle \mid \varphi_e(n)\downarrow\}$ is not computable.

The Halting Set

Here's what we know:

- $\mathbb{K}$ is computably enumerable, so it's the domain of a PCF, and it's the range of a CF.
- $\overline{\mathbb{K}}$ is not computably enumerable. So there's no program that can "list out" the elements of $\overline{\mathbb{K}}$.

Exercises.

- Find a computable function $f(i, j)$ so that for each k,

$$\lim_{i \to \infty} f(i, j) = \text{ the } i\text{th member of } \mathbb{K}.$$

- Same thing for $\overline{\mathbb{K}}$.

Computable Sets

Indexed family of functions.

Definition. Let $A \subseteq \mathbb{N}$. The *characteristic function of* A is

$$\chi_A(n) = \begin{cases} 1 & n \in A \\ 0 & n \notin A \end{cases}$$

Definition. A set of natural numbers is *computable* if its characteristic function is a computable function.

Exercise.
Let $h(j,k)$ be a total computable function. Define, for each j, a new function f_j by setting

$$f_j(k) = h(j,k).$$

Find a computable set so that for each j, $f_j \neq \chi_A$.

HINT: Diagonalization!

Indexed families of sets

An indexed family of sets $\{A_i \mid i \in \mathbb{N}\} \subseteq CE$ is computably indexable there is a computable function that, given i as input, outputs a program number for set A_i.

Example. Let $B_n = \{n\}$.

Example. Let $A_i = \{n \in \mathbb{N} \mid n \neq i\}$.

Indexed families of sets

Example. Let $D = \{d_0, d_1, \ldots, d_k\}$ be a non-empty finite set. Let $i_D = \prod_j p_{d_j} - 1$, where p_n denotes the nth prime.

Let A_i be the finite set encoded by i as in the scheme above, for $i > 0$.
Let $A_0 = \emptyset$.

Find a computable function that, for each i, outputs an index for A_i.

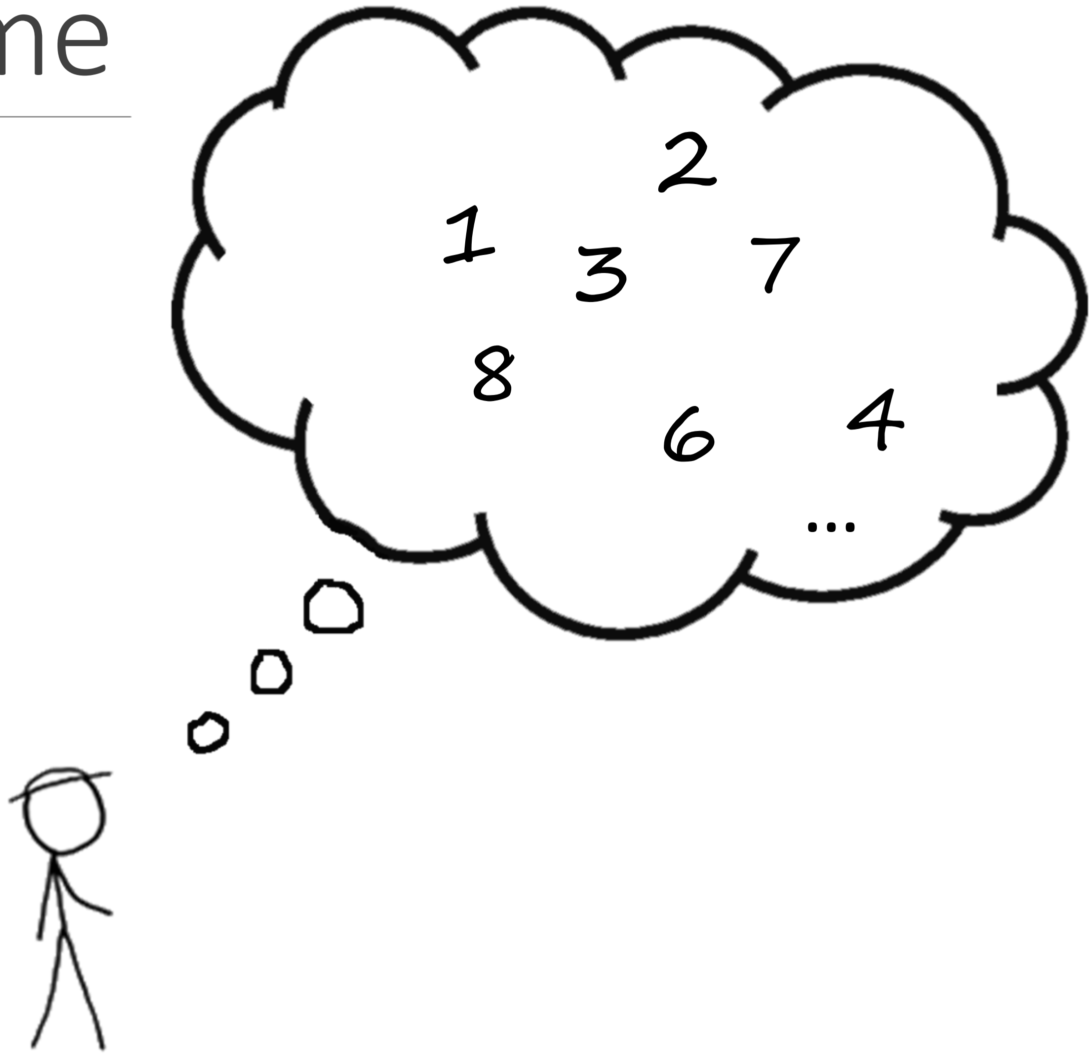
Back to our guessing game

I am thinking of a set. Can you guess what it is?

Our strategy was to guess "The set of all positive integers EXCEPT the smallest one we haven't seen."

1. INPUT $\langle n, t \rangle$.
2. Decode t into an n-tuple.
3. Let x be the smallest positive integer not appearing in the n-tuple.
4. Set e to be the number of a program with domain $\{n \mid n \neq x\}$.
5. OUTPUT e.

The grammar our learner conjectures is a program number.



Learners

"Consider a child learning a language. At any given moment the child has been exposed to only finitely many sentences. Yet he or she is typically willing to conjecture grammars for infinite languages. Within the identification paradigm the disposition to convert finite evidence into hypotheses about potentially infinite languages is the essential feature of a learner. More generally, the relation between finite evidential states and infinite languages is at the heart of inductive inference and learning theory."

Systems that Learn, by Osherson, Stob, and Weinstein

Identification: Basic concepts

Definitions.

1. A *language* is a c.e. set.
2. A *text* is an infinite sequence of natural numbers.
3. The set of numbers appearing in text t is denoted $rng(t)$.
4. We say t *is a text for language* L if $rng(t) = L$.
5. The set of all possible texts is denoted $\mathcal{T}$.

Identification: Basic concepts

- $t = (0, 0, 2, 2, 4, 4, 6, 6, \ldots)$ is a text for $L = \{2n \mid n \in \mathbb{N}\}$.
- So is $s = (2, 0, 6, 4, 10, 8, \ldots)$.
- If L has at least two elements, there are infinitely many texts for L.
- If L is a singleton, there is only one text for L.
- There are no texts for the empty language.

Notations: Let $t \in \mathcal{T}$ and $n \in \mathbb{N}$.

- $\overline{t_n}$ denotes the first n members of t.
- t_n is the nth member of t.

Identification: Basic concepts

We think of the text as the presentation of the language to the learner. At any given time, the learner will have seen only a finite initial sequence from the text.

- The set of all finite sequences is denoted SEQ, and can be thought of as the set of all possible evidential states a learner might see.
- We will usually use lower case greek letters σ, τ, etc. to denote elements of SEQ.
- The length of $\sigma \in SEQ$ is denoted $lh(\sigma)$, and the *set* consisting of the members of σ is $rng(\sigma)$.
- For $\sigma \in SEQ$ and $t \in \mathcal{T}$, we say $\sigma \in t$ if and only if $\sigma = \overline{t_{lh(\sigma)}}$.

Identification: Basic concepts

- Note that we can use our coding techniques (based on FTA) to encode finite strings as natural numbers, and so can identify SEQ with $\mathbb{N}$:

$$\langle \sigma \rangle = \langle lh(\sigma), \langle \sigma_1 \langle \sigma_2, \ldots \rangle \rangle \rangle$$

Now we are able to say what a *learner* really is in this paradigm. Informally, a learner converts a finite sequence of data into a conjecture. Formally...

Definition. A *learning function* is an element $\mathcal{F}$, i.e., a function from $\mathbb{N}$ to $\mathbb{N}$, where the domain is typically thought of as SEQ. It may be partial or total, computable or not.

Back to our guessing game

I am thinking of a set. Can you guess what it is?

Our strategy was to guess "The set of all positive integers EXCEPT the smallest one we haven't seen."

1. INPUT $\langle n, t \rangle$.
2. Decode t into an n-tuple.
3. Let x be the smallest positive integer not appearing in the n-tuple.
4. Set e to be the number of a program with domain $\{n \mid n \neq x\}$.
5. OUTPUT e.

The grammar our learner conjectures is a program number.