Anomalous Vacillatory Learning

Achilles Beros

Department of Mathematics,
UW - Madison

November 15, 2012

Background

Learning Theory is concerned with the identification of collections of *c.e.* sets by computable functions from enumerations of the sets.

Learning Theory is concerned with the identification of collections of $c.e.$ sets by computable functions from enumerations of the sets.

A first definition.

- A machine, M, identifies an enumeration, $(a_n)_{n\in\mathbb{N}}$, for a set A if $\lim_{i\to\infty} M(a_0a_1\dots a_i) = h$ and $W_h = A$.
- M learns a set, A, if it identifies every enumeration.
- M learns a family of sets if it learns each set in the family.

Background

Learning Theory is concerned with the identification of collections of *c.e.* sets by computable functions from enumerations of the sets.

A first definition.

- A machine, M, identifies an enumeration, $(a_n)_{n\in\mathbb{N}}$, for a set A if $\lim_{i\to\infty} M(a_0a_1\ldots a_i) = h$ and $W_h = A$.
- M learns a set, A, if it identifies every enumeration.
- M learns a family of sets if it learns each set in the family.

This is known as TxtEx-learning, or learning in the limit.

Learning Theory is concerned with the identification of collections of *c.e.* sets by computable functions from enumerations of the sets.

A first definition.

- A machine, M, identifies an enumeration, $(a_n)_{n\in\mathbb{N}}$, for a set A if $\lim_{i\to\infty} M(a_0 a_1 \dots a_i) = h$ and $W_h = A$.
- M learns a set, A, if it identifies every enumeration.
- M learns a family of sets if it learns each set in the family.

This is known as TxtEx-learning, or learning in the limit.

`input data stream:`	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	...
`learning machine:`	$\downarrow$		$\downarrow$				$\downarrow$	$\downarrow$			...
`hypothesis stream:`	h_0		h_1				h_2	h_3			...

Background

Learning Theory is concerned with the identification of collections of *c.e.* sets by computable functions from enumerations of the sets.

A first definition.

- A machine, M, identifies an enumeration, $(a_n)_{n\in\mathbb{N}}$, for a set A if $\lim_{i\to\infty} M(a_0 a_1 \dots a_i) = h$ and $W_h = A$.
- M learns a set, A, if it identifies every enumeration.
- M learns a family of sets if it learns each set in the family.

This is known as TxtEx-learning, or learning in the limit.

input data stream:	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	...
learning machine:	$\downarrow$		$\downarrow$				$\downarrow$	$\downarrow$			...
hypothesis stream:	h_0		h_1				h_2	h_3			...

Examples

1. $\{A\}$ is learnable for any $c.e.$ set A.
2. $\{F : F \text{ is a finite set}\}$ is learnable by $M(\sigma) = e$ where $W_e = \text{content}(\sigma)$.
3. $\{F : F \text{ is a finite set}\} \cup \{\mathbb{N}\}$ is not learnable.
4. $\{A \cup \{x\} : x \in \mathbb{N}\}$ is not learnable for any non-computable, $c.e.$ set A.

Examples

1. $\{A\}$ is learnable for any $c.e.$ set A.
2. $\{F : F \text{ is a finite set}\}$ is learnable by $M(\sigma) = e$ where $W_e = \text{content}(\sigma)$.
3. $\{F : F \text{ is a finite set}\} \cup \{\mathbb{N}\}$ is not learnable.
4. $\{A \cup \{x\} : x \in \mathbb{N}\}$ is not learnable for any non-computable, $c.e.$ set A.

Learning with errors is called anomalous learning.

We consider two types of anomalous vacillatory learning, TxtFex^*_* and $\mathrm{TxtFext}^*_*$.

We consider two types of anomalous vacillatory learning, TxtFex^*_* and $\mathrm{TxtFext}^*_*$.

TxtFex^*_*

M TxtFex^*_*-identifies a text T if, and only if,

$$(\exists S \text{ finite})(\forall^\infty n)(\forall a \in S)(M(T[n]) \in S \wedge W_a =^* \mathrm{content}(T)).$$

We consider two types of anomalous vacillatory learning, $\mathbf{TxtFex}_*^*$ and $\mathbf{TxtFext}_*^*$.

$\mathbf{TxtFex}_*^*$

M $\mathbf{TxtFex}_*^*$-identifies a text T if, and only if,

$$(\exists S \text{ finite})(\forall^\infty n)(\forall a \in S)(M(T[n]) \in S \wedge W_a =^* \text{content}(T)).$$

$\mathbf{TxtFext}_*^*$

M $\mathbf{TxtFext}_*^*$-identifies a text T if, and only if,

$$(\exists S \text{ finite})(\forall^\infty n)(\forall a, b \in S)(M(T[n]) \in S \wedge W_a = W_b =^* \text{content}(T)).$$

Superficially, the hypothesis streams look quite similar:

Superficially, the hypothesis streams look quite similar:

$$\text{TxtFex}^*_*:\ h_0\ h_1\ h_2\ h_3\ h_4\ h_5\ h_6\ h_5\ h_6\ h_4\ h_5 \ldots$$

$$\text{TxtFext}^*_*:\ k_0\ k_1\ k_2\ k_3\ k_4\ k_5\ k_6\ k_5\ k_6\ k_4\ k_5 \ldots$$

Colored hypotheses are repeated infinitely often

Superficially, the hypothesis streams look quite similar:

$$\mathrm{TxtFex}^*_*:\ h_0\ h_1\ h_2\ h_3\ h_4\ h_5\ h_6\ h_5\ h_6\ h_4\ h_5 \ldots$$
$$\mathrm{TxtFext}^*_*:\ k_0\ k_1\ k_2\ k_3\ k_4\ k_5\ k_6\ k_5\ k_6\ k_4\ k_5 \ldots$$

Colored hypotheses are repeated infinitely often

$$\mathrm{TxtFex}^*_*:\ h_0\ h_1\ h_2\ h_3\ h_4\ h_5\ h_6\ h_5\ h_6\ h_4\ h_5 \ldots$$
$$\mathrm{TxtFext}^*_*:\ k_0\ k_1\ k_2\ k_3\ k_4\ k_5\ k_6\ k_5\ k_6\ k_4\ k_5 \ldots$$

Colored hypotheses are repeated infinitely often and hypotheses of different colors code different sets.

Question

- In 1986, Osherson, Stob and Weinstein asked whether TxtFex^*_* and TxtFext^*_* are equivalent.

Question

- In 1986, Osherson, Stob and Weinstein asked whether TxtFex_*^* and $\mathrm{TxtFext}_*^*$ are equivalent.
- In 1994, Fulk, Jain and Osherson proved that $(\forall j \in \mathbb{N})(\mathrm{TxtFex}_*^j \subseteq \mathrm{TxtFext}_*^*)$.

Question

- In 1986, Osherson, Stob and Weinstein asked whether TxtFex^*_* and $\mathrm{TxtFext}^*_*$ are equivalent.
- In 1994, Fulk, Jain and Osherson proved that $(\forall j \in \mathbb{N})(\mathrm{TxtFex}^j_* \subseteq \mathrm{TxtFext}^*_*)$.
- Earlier this year, I proved $\mathrm{TxtFex}^*_2 \neq \mathrm{TxtFext}^*_*$. This proves $\mathrm{TxtFex}^*_* \neq \mathrm{TxtFext}^*_*$ in the strongest possible way.

Theorem

There is a u.c.e. family that is $\mathbf{TxtFex}_2^*$*-learnable, but not* $\mathbf{TxtFext}_*^*$*-learnable.*

Theorem

There is a u.c.e. family that is $TxtFex^*_2$-learnable, but not $TxtFext^*_*$-learnable.

In particular,

Corollary

$TxtFex^*_* \neq TxtFext^*_*$

Theorem

There is a u.c.e. family that is $TxtFex^*_2$-learnable, but not $TxtFext^*_*$-learnable.

In particular,

Corollary

$TxtFex^*_* \neq TxtFext^*_*$

The proof started as an infinite-injury priority argument, but the tree collapsed.

Theorem
There is a u.c.e. family that is $TxtFex^*_2$-learnable, but not $TxtFext^*_*$-learnable.

In particular,

Corollary
$TxtFex^*_* \neq TxtFext^*_*$

The proof started as an infinite-injury priority argument, but the tree collapsed.

We diagonalize against every possible machine by forcing machines to commit to a finite number of hypotheses.

Proof.

- A single step of the argument diagonalizes against a machine, M_e, by building a family, $\mathcal{L}_e$.

Proof.

- A single step of the argument diagonalizes against a machine, M_e, by building a family, $\mathcal{L}_e$.
- Let $L_e = \{e, e+1, e+2, \ldots\}$

Proof.

- A single step of the argument diagonalizes against a machine, M_e, by building a family, $\mathcal{L}_e$.
- Let $L_e = \{e, e+1, e+2, \ldots\}$
- We consider the tree of strings whose content is contained in L_e. All cones referred to subsequently will be subsets of this tree.

Proof.

- A single step of the argument diagonalizes against a machine, M_e, by building a family, $\mathcal{L}_e$.
- Let $L_e = \{e, e+1, e+2, \dots\}$
- We consider the tree of strings whose content is contained in L_e. All cones referred to subsequently will be subsets of this tree.
- Define σ to be an (e, k)-stabilizing sequence iff $[e, e+k] \subseteq \text{content}(\sigma) \subset L_e$ and for τ in the cone below σ
  1. $M_e(\tau) \leq |\sigma|$
  2. $W_{M_e(\sigma)} \cap [0, k) = W_{M_e(\tau)} \cap [0, k)$

Proof.

- A single step of the argument diagonalizes against a machine, M_e, by building a family, $\mathcal{L}_e$.
- Let $L_e = \{e, e+1, e+2, \ldots\}$
- We consider the tree of strings whose content is contained in L_e. All cones referred to subsequently will be subsets of this tree.
- Define σ to be an (e,k)-stabilizing sequence iff $[e, e+k] \subseteq \text{content}(\sigma) \subset L_e$ and for τ in the cone below σ
  1. $M_e(\tau) \leq |\sigma|$
  2. $W_{M_e(\sigma)} \cap [0,k) = W_{M_e(\tau)} \cap [0,k)$
- "σ is not an (e,k)-stabilizing sequence" is Σ^0_1.

- If there is an (e, n)-stabilizing sequence, there is a computable sequence of strings that converges to an (e, n)-stabilizing sequence in the limit.

The Proof

- If there is an (e, n)-stabilizing sequence, there is a computable sequence of strings that converges to an (e, n)-stabilizing sequence in the limit.
- More generally, define $\sigma_{e,k,s}$ to be an array of strings, with $\lim_{s\to\infty} \sigma_{e,k,s} = \sigma_{e,k}$, if it exists, such that:
  - $\sigma_{e,k,s} \prec \sigma_{e,k+1,s}$ for all $k, s \in \mathbb{N}$.
  - If $\sigma_{e,0}, \ldots \sigma_{e,n}$ are defined and there is an $(e, n+1)$-stabilizing sequence extending $\sigma_{e,n}$, then $(\sigma_{e,n+1,s})_{s\in\mathbb{N}}$ converges to such a string.

- If there is an (e, n)-stabilizing sequence, there is a computable sequence of strings that converges to an (e, n)-stabilizing sequence in the limit.
- More generally, define $\sigma_{e,k,s}$ to be an array of strings, with $\lim_{s\to\infty} \sigma_{e,k,s} = \sigma_{e,k}$, if it exists, such that:
  - $\sigma_{e,k,s} \prec \sigma_{e,k+1,s}$ for all $k, s \in \mathbb{N}$.
  - If $\sigma_{e,0}, \ldots \sigma_{e,n}$ are defined and there is an $(e, n+1)$-stabilizing sequence extending $\sigma_{e,n}$, then $(\sigma_{e,n+1,s})_{s\in\mathbb{N}}$ converges to such a string.
- Let $a_{e,k}$ be the least even number such that $\sigma_{e,h,s} = \sigma_{e,h,s+1}$ for $h \leq k$ and $s \geq a_{e,k}$. Define $b_{e,k} = a_{e,k} + 1$.

The Proof

- Define $R_e = L_e \setminus \{a_{e,i} : i \in \mathbb{N}\}$ and $R_e^* = L_e \setminus \{b_{e,i} : i \in \mathbb{N}\}$, both of which are $c.e.$

- Define $R_e = L_e \setminus \{a_{e,i} : i \in \mathbb{N}\}$ and $R_e^* = L_e \setminus \{b_{e,i} : i \in \mathbb{N}\}$, both of which are $c.e.$
- Let $\mathcal{L}_e = \{R_e \cup (F \cap L_e) : F \text{ is a finite set}\} \cup \{R_e^* \cup (F \cap L_e) : F \text{ is a finite set}\}$.

- Define $R_e = L_e \setminus \{a_{e,i} : i \in \mathbb{N}\}$ and $R_e^* = L_e \setminus \{b_{e,i} : i \in \mathbb{N}\}$, both of which are $c.e.$
- Let $\mathcal{L}_e = \{R_e \cup (F \cap L_e) : F \text{ is a finite set}\} \cup \{R_e^* \cup (F \cap L_e) : F \text{ is a finite set}\}$.
- We must prove two claims:
  1. $\mathcal{L}_e$ is not TxtFext_*^*-learnable by the fixed machine, M_e.
  2. $\bigcup_{e \in \mathbb{N}} \mathcal{L}_e$ is TxtFex_2^*-learnable.

1. M_e does not $\mathrm{TxtFext}^*_*$-learn $\mathcal{L}_e$

- Suppose there is a minimal $k \neq 0$ such that $\sigma_{e,k}$ is undefined.

1. M_e does not $\mathrm{TxtFext}^*_*$-learn $\mathcal{L}_e$

- Suppose there is a minimal $k \neq 0$ such that $\sigma_{e,k}$ is undefined.
  - There is no $\sigma \succ \sigma_{e,k-1}$ such that, in the cone below σ, all hypotheses code sets that agree on $[0, k)$.

1. M_e does not $\mathrm{TxtFext}^*_*$-learn $\mathcal{L}_e$

- Suppose there is a minimal $k \neq 0$ such that $\sigma_{e,k}$ is undefined.
  - There is no $\sigma \succ \sigma_{e,k-1}$ such that, in the cone below σ, all hypotheses code sets that agree on $[0, k)$.
  - Inductively build an enumeration of L_e on which M_e infinitely often outputs codes for two sets that are not equal.

1. M_e does not TxtFext^*_*-learn $\mathcal{L}_e$

- Suppose there is a minimal $k \neq 0$ such that $\sigma_{e,k}$ is undefined.
  - There is no $\sigma \succ \sigma_{e,k-1}$ such that, in the cone below σ, all hypotheses code sets that agree on $[0, k)$.
  - Inductively build an enumeration of L_e on which M_e infinitely often outputs codes for two sets that are not equal.
- Suppose $\sigma_{e,0}$ is undefined.
  - If possible, build an enumeration as above.
  - If not, then build an enumeration on which M_e nevers settles upon a finite list of hypotheses.

- Suppose $\sigma_{e,k}$ is defined for all $k \in \mathbb{N}$.

- Suppose $\sigma_{e,k}$ is defined for all $k \in \mathbb{N}$.
  - R_e and R_e^* are both coinfinite and have infinite symmetric difference.
  - By the definition of $\sigma_{e,0}$, there is a finite list, $h_0, h_1, \ldots, h_n$, of distinct hypotheses M_e outputs on extensions of $\sigma_{e,0}$.

- Suppose $\sigma_{e,k}$ is defined for all $k \in \mathbb{N}$.
  - R_e and R_e^* are both coinfinite and have infinite symmetric difference.
  - By the definition of $\sigma_{e,0}$, there is a finite list, $h_0, h_1, \ldots, h_n$, of distinct hypotheses M_e outputs on extensions of $\sigma_{e,0}$.
  - Pick k large enough so that $(\forall i, j \leq n)(\exists x \leq k)(W_{h_i} \neq W_{h_j} \rightarrow x \in W_{h_i} \triangle W_{h_j})$.
  - All hypotheses made on extensions of $\sigma_{e,k}$ contained in L_e must agree up to $k-1$, thus be equal.

- Suppose $\sigma_{e,k}$ is defined for all $k \in \mathbb{N}$.
  - R_e and R_e^* are both coinfinite and have infinite symmetric difference.
  - By the definition of $\sigma_{e,0}$, there is a finite list, $h_0, h_1, \ldots, h_n$, of distinct hypotheses M_e outputs on extensions of $\sigma_{e,0}$.
  - Pick k large enough so that $(\forall i, j \leq n)(\exists x \leq k)(W_{h_i} \neq W_{h_j} \rightarrow x \in W_{h_i} \triangle W_{h_j})$.
  - All hypotheses made on extensions of $\sigma_{e,k}$ contained in L_e must agree up to $k-1$, thus be equal.
  - $A = \text{content}(\sigma_{e,k}) \cup R_e$ and $B = \text{content}(\sigma_{e,k}) \cup R_e^*$ extend $\sigma_{e,k}$ and have infinite symmetric difference.

2. $\bigcup_{e\in\mathbb{N}} \mathcal{L}_e$ is TxtFex_2^*-learnable

2. $\bigcup_{e\in\mathbb{N}} \mathcal{L}_e$ is TxtFex_2^*-learnable

- Let x_e and x_e^* code R_e and R_e^*, respectively.
- R_e is co-even and R_e^* is co-odd above e, hence the sets are recognizable by the types of numbers they do not contain.

2. $\bigcup_{e\in\mathbb{N}} \mathcal{L}_e$ is TxtFex_2^*-learnable

- Let x_e and x_e^* code R_e and R_e^*, respectively.
- R_e is co-even and R_e^* is co-odd above e, hence the sets are recognizable by the types of numbers they do not contain.
- For an input string, σ, we define the following:
  - $m_\sigma = \min(\text{content}(\sigma))$.
  - $n_\sigma = \min(\{y > m_\sigma : y \notin \text{content}(\sigma)\})$.

2. $\bigcup_{e\in\mathbb{N}} \mathcal{L}_e$ is TxtFex_2^*-learnable

- Let x_e and x_e^* code R_e and R_e^*, respectively.
- R_e is co-even and R_e^* is co-odd above e, hence the sets are recognizable by the types of numbers they do not contain.
- For an input string, σ, we define the following:
  - $m_\sigma = \min(\text{content}(\sigma))$.
  - $n_\sigma = \min(\{y > m_\sigma : y \notin \text{content}(\sigma)\})$.
- Define a learning machine, M, by

$$M(\sigma) = \begin{cases} x_e & e = m_\sigma \wedge (n_\sigma \text{ is even}) \\ x_e^* & e = m_\sigma \wedge (n_\sigma \text{ is odd}) \\ 0 & \text{otherwise} \end{cases}$$

- M recieves an enumeration for $R_e \cup F$.
- If R_e is cofinite, then both x_e and x_e^* are correct hypotheses.
- If R_e is coinfinite, then the least element of $L_e \setminus (R_e \cup F)$ is an even number.
- Cofinitely, M will output the hypothesis x_e.

□

Other Learning Criteria

TxtFin

Fix a symbol "?". M TxtFin-identifies a text T if, and only if,
$\exists n \forall n' < n (M(T[n']) = ? \wedge M(T[n]) \neq ? \wedge W_{M(T[n])} = \text{content}(T))$.

TxtEx

M TxtEx-identifies a text T if, and only if,
$\exists n (\lim_{i \to \infty} M(T[i]) \to n \wedge W_n = \text{content}(T))$.

TxtBC

M TxtBC-identifies a text T if, and only if,
$\exists n \forall i > n (W_{M(T[i])} = \text{content}(T))$.

TxtEx*

M TxtEx*-identifies a text T if, and only if,
$\exists n (\lim_{i \to \infty} M(T[i]) \to n \wedge W_n =^* \text{content}(T))$.

Index Sets

1. Let FINL denote the index set of all Σ^0_1 codes for *u.c.e.* families such that $e \in$ FINL if, and only if, e codes a TxtFin-learnable family.
2. Let EXL denote the index set of all Σ^0_1 codes for *u.c.e.* families such that $e \in$ EXL if, and only if, e codes a TxtEx-learnable family.
3. Let BCL denote the index set of all Σ^0_1 codes for *u.c.e.* families such that $e \in$ BCL if, and only if, e codes a TxtBC-learnable family.
4. Let EXL* denote the index set of all Σ^0_1 codes for *u.c.e.* families such that $e \in$ EXL* if, and only if, e codes a TxtEx*-learnable family.

Theorem

FINL is Σ^0_3*-complete*

Theorem

EXL is Σ^0_4*-complete*

Theorem

BCL is Σ^0_5*-complete*

Theorem

EXL^* *is* Σ^0_5*-complete*

Thank You