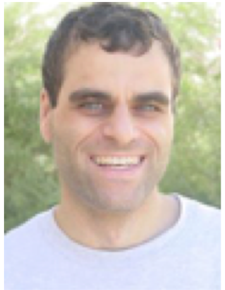
Time-Space Tradeoffs for Two-Pass Learning

Sumegha Garg (Princeton)

Joint Work with Ran Raz (Princeton) and Avishay Tal (UC Berkeley)



[Shamir 14], [Steinhardt-Valiant-Wager 15]

Initiated a study of memory-samples lower bounds for learning

Can one prove unconditional lower bounds on the number of samples needed for learning under memory constraints?

(when samples are viewed one by one)

(also known as online learning)

When two-passes are allowed?

Can one prove unconditional lower bounds on the number of samples needed for learning under memory constraints,

and when learner is allowed to go over the stream of samples twice?

(in the same order)

Toy-Example: Parity Learning

Parity Learning

$x \in_R \{0,1\}^n$ is unknown

A learner tries to learn x from

$(a_1, b_1), (a_2, b_2), \dots, (a_m, b_m)$, where $\forall\ t$,

$a_t \in_R \{0,1\}^n$ and $b_t =< a_t, x >$ (inner product mod 2)

In other words, learner gets random linear equations in $x_1, x_2, .., x_n$, one by one, and need to solve them

Parity Learners

- Solve independent linear equations (Gaussian Elimination)
  - $O(n)$ samples and $O(n^2)$ memory
- Try all possibilities of x
  - $O(n)$ memory but exponential number of samples

Parity Learning (Two-pass)

$x \in_R \{0,1\}^n$ is unknown

A learner tries to learn x from

$(a_1, b_1), (a_2, b_2), \dots, (a_m, b_m), (a_1, b_1), (a_2, b_2), \dots, (a_m, b_m)$, where $\forall\ t$,

$a_t \in_R \{0,1\}^n$ and $b_t =< a_t, x >$ (inner product mod 2)

Raz's Breakthough '16 (One-pass)

Any algorithm for parity learning of size n requires either $\Omega(n^2)$ memory bits or an exponential number of samples

Conjectured by:

Steinhardt, Valiant and Wager [2015]

Subsequent Results (One-pass)

[Kol-Raz-Tal ‘17]: Generalization to sparse parities

[Raz’17, Moshkovitz-Moshkovitz’17, Moshkovitz- Tishby’17, Moshkovitz-Moshkovitz’18, Garg-Raz-Tal’18, Beame-Gharan-Yang’18]:
Generalization to larger class of problems

[Sharan-Sidford-Valiant’19]: Generalization to real-valued learning

Related Results (Multiple-pass)

[Dagan-Shamir’18, Assadi-Chen-Khanna’19,...]: Uses communication complexity

(Quite different technique, at most polynomial bound on the number of samples)

Motivation

Learning Theory, Bounded Storage Cryptography, Complexity Theory

With [Barrington'89], proving super-polynomial lower bounds on the time needed for computing a function, by a branching program of width 5, with polynomially many passes over the input, would imply super-polynomial lower bounds for formula size

Technically Challenging: previous techniques are heavily based on the fact that in the one-pass case all the samples are independent

Our Result

Our Result for Parity Learning

Any two-pass algorithm for parity learning of size n requires either $\Omega(n^{1.5})$ memory bits or $2^{\Omega(\sqrt{n})}$ number of samples

(no matching upper bound)

Learning Problem as a Matrix

A, X : finite sets, $M: A \times X \rightarrow \{-1,1\}$: a matrix

$x \in_R X$ is unknown. A learner tries to learn x from a stream

$(a_1, b_1), (a_2, b_2), \dots, (a_m, b_m), (a_1, b_1), (a_2, b_2), \dots, (a_m, b_m)$, where $\forall t$:

$a_t \in_R A$ and

$b_t = M(a_t, x)$

X : concept class $= \{0,1\}^n$

A : possible samples $= \{0,1\}^{n'}$

Generalized Result

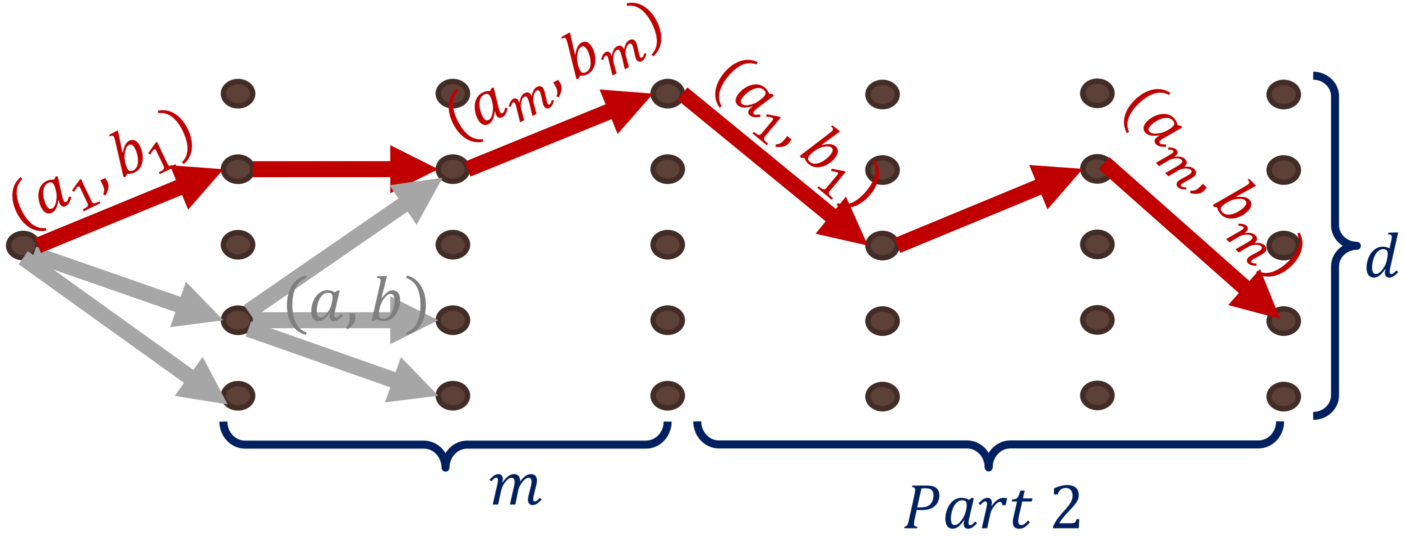
Assume that any submatrix of M of at least $2^{-k}|A|$ rows and at least $2^{-\ell}|X|$ columns, has a bias of at most 2^{-r}. Then:

Any two-pass algorithm requires either $\Omega(k \cdot \min\{k, \sqrt{l}\})$ memory bits or $2^{\Omega(\min\{k,\sqrt{l},r\})}$ samples

In contrast, [GRT'18] proved

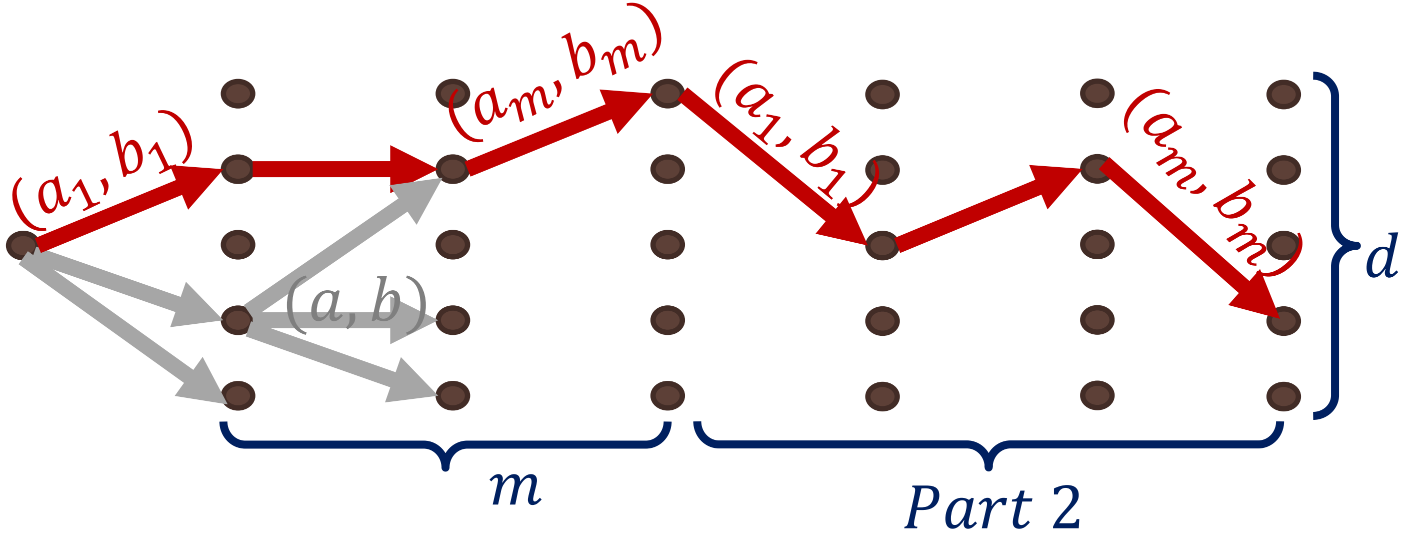
Any one-pass algorithm requires either $\Omega(k \cdot l)$ memory bits or $2^{\Omega(r)}$ samples

Branching Program (length m, width d, 2-pass)



Each layer represents a time step. Each vertex represents a memory state of the learner ($d = 2^{memory}$). Each non-leaf vertex has $2^{n'+1}$ outgoing edges, one for each $(a, b) \in \{0,1\}^{n'} \times \{-1,1\}$

Branching Program (length m, width d, 2-pass)



The samples $(a_1, b_1), .., (a_m, b_m), (a_1, b_1), .., (a_m, b_m)$ define a computation-path. Each vertex v in the last layer is labeled by $\hat{x}_v \in \{0,1\}^n$. The output is the label $\hat{x}_v$ of the vertex reached by the path

Brief Overview of One-Pass Lower Bound [GRT'18]

$\mathrm{P}_{x|v}$ = distribution of x conditioned on the event that the computation-path reaches v

Significant vertices: v s.t. $||\mathrm{P}_{x|v}||_2 \geq 2^l \cdot 2^{-n}$

$Pr(v)$ = probability that the path reaches v

GRT proves: If v is significant, $Pr(v) \leq 2^{-\Omega(k \cdot l)}$

Hence, there are at least $2^{\Omega(k \cdot l)}$ significant vertices to output correct answer with high probability

Brief Overview of One-Pass Lower Bound

$\mathrm{P}_{x|v}$ = distribution of x conditioned on the event that the computation-path reaches v

$Pr(v)$ = probability that the path reaches v under

T = same as the computational path, but stops when "atypical" things happen (traversing a bad edge and …)

Bad edges: a s.t. $|(M \cdot \mathrm{P}_{x|v})(a)| \geq 2^{-r}$

$Pr(T\ stops)$ is exp small (uses $a \in_R \{0,1\}^n$!)

Difficulties for Two-Passes (1)

$\mathrm{P}_{a|v} \neq Uniform_{\{0,1\}^n}$ for v in Part-2

For e.g., BP remembers a_1. Therefore, probability of traversing a "bad edge" may not be small

Bad edges: $|(M \cdot \mathrm{P}_{x|v})(a)| \geq 2^{-r}$ (gives too much information about x)

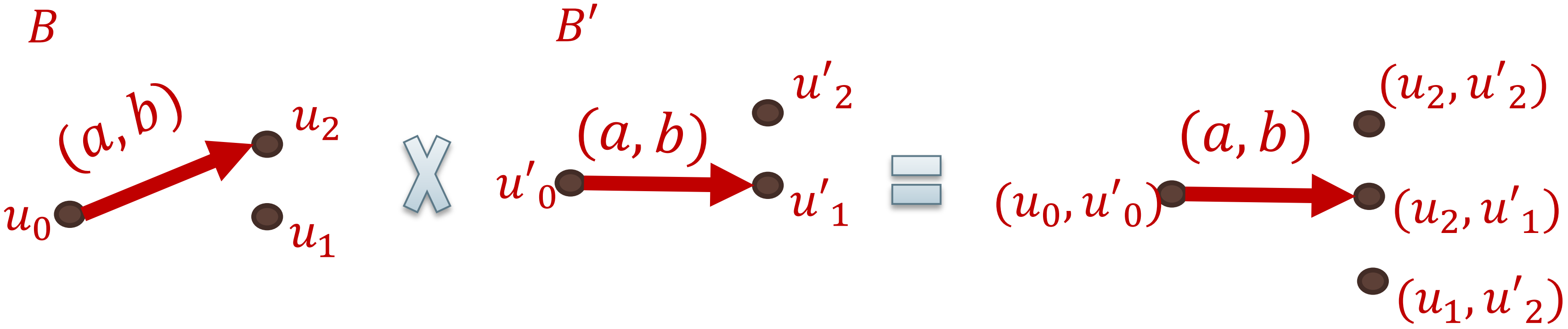
Save: can't remember too many as. New stopping rules!

Difficulties for Two-Passes (2)

Proving -- if v is significant, then $Pr(v) \leq 2^{-\Omega(k \cdot l)}$ -- uses $a \in_R \{0,1\}^n$ along with extractor property

Save: work on product of 2 parts which is read-once. New stopping rules!

Product of 2 Parts (length m, width d^2, 1-pass)



Let v_0 be start vertex of 2-pass BP. v_1, v_2 be the vertices reached in the end of Part-1 and 2 respectively. Then

$$v_0 \to v_1 \to v_2 \equiv (v_0, v_1) \to (v_1, v_2)$$

Proof Outline: Stopping Rules for Product

Significant vertices: (v, v') s.t. $||\mathrm{P}_{x|(v_0,v_1)\to(v,v')}||_2 \geq 2^l \cdot 2^{-n}$

Bad edges: a s.t. $|(M \cdot \mathrm{P}_{x|(v_0,v_1)\to(v,v')})(a)| \geq 2^{-r}$

High-probability edges: a s.t. $\Pr[a|v_0 \to v \to v_1 \to v'] \geq 2^k \cdot 2^{-n}$

….

Stop at bad edges unless high-probability edges

unless they are very bad

Proof Outline: Stopping Rules for Product

Conditioned on $v_0 \to v \to v_1 \to v'$, $Pr(stops)$ is small (1/100)

$$v_0 \to v \to v_1 \to v' \neq (v_0, v_1) \to (v, v')$$

Proved using single-pass result as a subroutine

Open Problems

- Generalize to multiple-passes
- Better lower-bounds for two-pass
- Non-trivial upper bounds for constant, linear passes

Thank You!

Anyone wants a second pass?