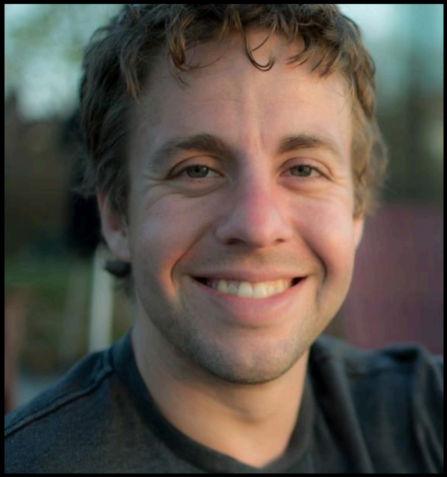
Charting the Landscape of Memory/Data Tradeoffs in Continuous Optimization: A Survey of Open Problems



Annie Marsden **Vatsal Sharan** **Aaron Sidford**

How do ***memory constraints*** influence the speed of learning/optimization?

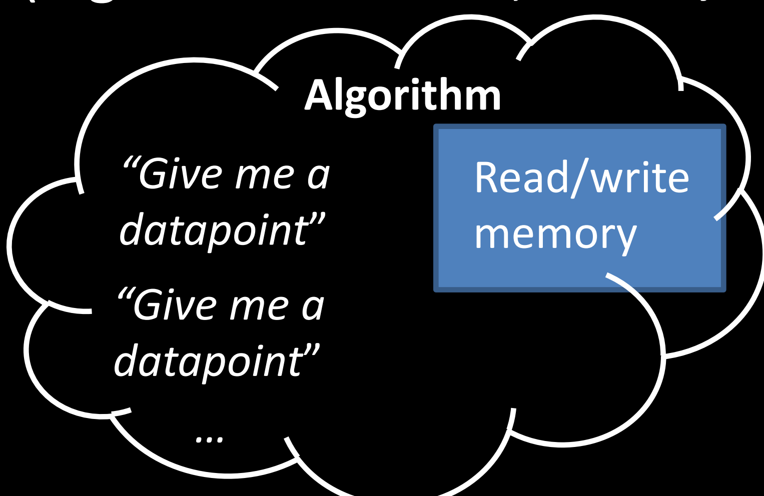
Today: Linear Regression *min x'Ax – bx*

(or solving Ax = b)

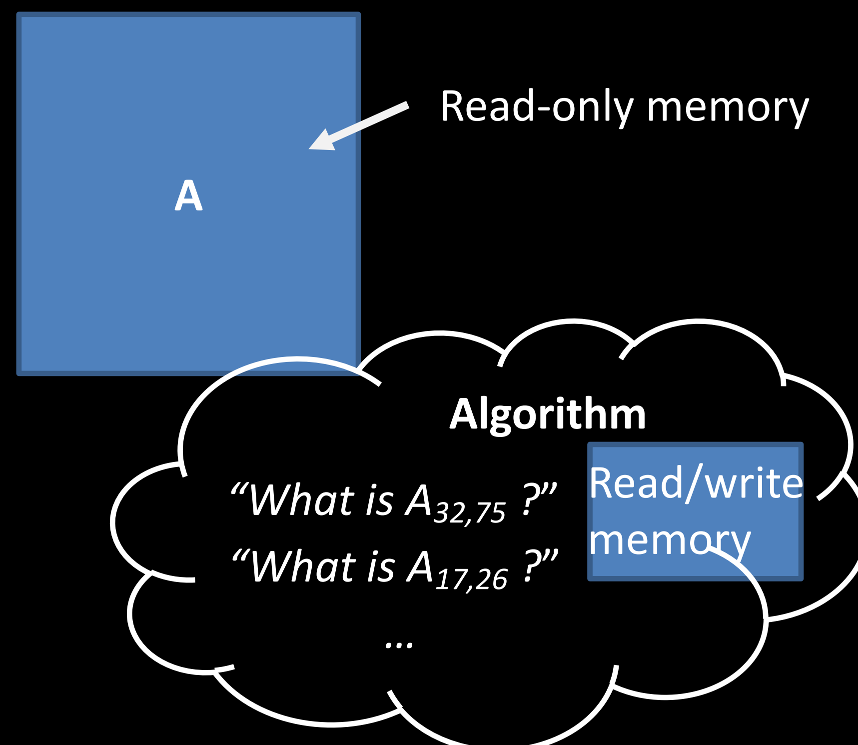
Access to i.i.d. data samples

Distribution over (a,b) pairs

(e.g. a **<--** Gaussian, b = <a**,** x>)



"Cell-Probe" Model



	i.i.d. data samples	"Cell-Probe" Model
Discrete setting Solve Ax=b over a finite field	Unknown $x \in \{0,1\}^d$ chosen uniformly at random. Each datapoint: $(a_i, \ < a_i, x > \bmod 2)$, $a_i \in \{0,1\}^d$ chosen unif. rand Goal: find *x*	Same as i.i.d. data setting, but datapoints stored in read-only memory.
Continuous setting Solve Ax=b over $\mathbf{R}^d$ (or regression)	Unknown $x \in \mathrm{R}^d$ with $\|x\|=1$ chosen uniformly at random. Each datapoint: $(a_i, \ < a_i, x >)$, $a_i \in \mathrm{R}^d$ chosen e.g. from $N(0,I_d)$ [or some other distribution] Goal: approximate *x*	Same as i.i.d. data setting, but datapoints stored in read-only memory.

Many other interesting models…

No direct access to datapoints, but instead interact via specific types of oracles

- Statistical Query access
- Function evaluation queries
- Gradient queries
- Etc.

	i.i.d. data samples	"Cell-Probe" Model
Discrete setting Solve Ax=b over a finite field	Unknown $x \in \{0,1\}^d$ chosen uniformly at random. Each datapoint: $(a_i, \; < a_i, x > \bmod 2)$, $a_i \in \{0,1\}^d$ chosen unif. rand Goal: find *x*	Same as i.i.d. data setting, but datapoints stored in read-only memory.
Continuous setting Solve Ax=b over $\mathbf{R}^d$ (or regression)	Unknown $x \in \mathrm{R}^d$ with |*x*|=1 chosen uniformly at random. Each datapoint: $(a_i, \; < a_i, x > \; + noise)$ $a_i \in \mathrm{R}^d$ chosen e.g. from $N(0, I_d)$ [or some other distribution] Goal: approximate *x*	Same as i.i.d. data setting, but datapoints stored in read-only memory.

	i.i.d. data samples	"Cell-Probe" Model
Discrete setting Solve Ax=b over a finite field	Unknown $x \in \{0,1\}^d$ chosen uniformly at random. Each datapoint: $(a_i, \ < a_i, x > \bmod 2)$, $a_i \in \{0,1\}^d$ chosen unif. rand Goal: find *x*	Same as i.i.d. data setting, but datapoints stored in read-only memory.
Continuous setting Solve Ax=b over $\mathbf{R}^d$ (or regression)	Unknown $x \in \mathrm{R}^d$ with $\|x\|$=1 chosen uniformly at random. Each datapoint: $(a_i, \ < a_i, x > \ + noise)$ $a_i \in \mathrm{R}^d$ chosen e.g. from $N(0, I_d)$ [or some other distribution] Goal: approximate *x*	Same as i.i.d. data setting, but datapoints stored in read-only memory.

Unknown $x \in \{0,1\}^d$ chosen uniformly at random.

Given access to stream of examples $(a_i, < a_i, x > \bmod 2)$,

$a_i \in \{0,1\}^d$ chosen uniformly at random.

Gaussian elimination: O(d^2) memory , O(d) examples

Brute force guess/check: O(d) memory , $2^{O(d)}$ examples.

Conjecture [Steinhardt, Valiant, Wager'15]:

Any algorithm with <$d^2/4$ memory needs exponential number of samples to learn x.

Unknown $x \in \{0,1\}^d$ chosen uniformly at random.

Given access to stream of examples $(a_i, \ < a_i, x > \bmod 2)$,

$a_i \in \{0,1\}^d$ chosen uniformly at random.

Gaussian elimination: $O(d^2)$ memory , $O(d)$ examples

Brute force guess/check: $O(d)$ memory , $2^{O(d)}$ examples.

Thm [Raz'16]:

~~Conjecture [Steinhardt, Valiant, Wager'15]~~:

Any algorithm with $<d^2/\cancel{4}$ 20 memory needs exponential number of samples to learn x.

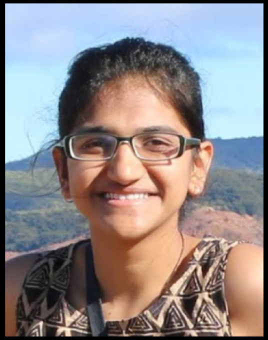
Subsequent work extended this to a broad class of learning problems over finite fields

Kol-Raz-Tal'17: sparse parities

Raz'17, Moshkovitz-Moshkovitz'17,18, Beame-Ovies Gharan-Yang'18, Garg-Raz-Tal'18: Large class of learning problems over finite fields satisfying combinatorial/mixing properties.

...and more recent papers

See Sumegha Garg's talk Thursday!!



Unknown $x \in \{0,1\}^d$ chosen uniformly at random.

Given access to stream of examples $(a_i, < a_i, x > \bmod 2)$,

$a_i \in \{0,1\}^d$ chosen uniformly at random.

Gaussian elimination: $O(d^2)$ memory , $O(d)$ examples

Brute force guess/check: $O(d)$ memory , $2^{O(d)}$ examples.

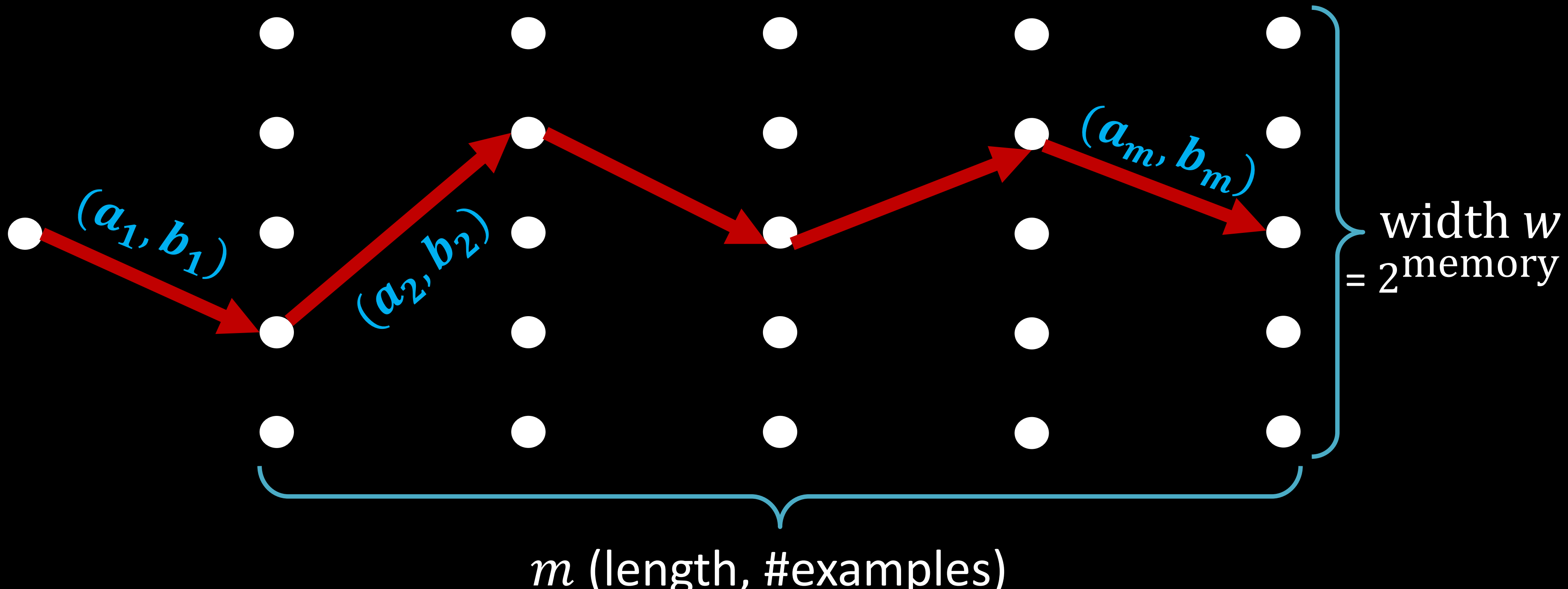
Thm [Raz'16]:

~~Conjecture [Steinhardt, Valiant, Wager'15]~~:

Any algorithm with $< d^2/\cancel{4}$ memory needs
20

exponential number of samples to learn x.

Branching program for learning



Each layer corresponds to a time step

Each vertex corresponds to a memory state

Every memory state has a transition function (an \`edge') which is a mapping from an example (a, b) to a vertex in the next layer.

	i.i.d. data samples	"Cell-Probe" Model
Discrete setting Solve Ax=b over a finite field	Unknown $x \in \{0,1\}^d$ chosen uniformly at random. Each datapoint: $(a_i, < a_i, x > \text{mod } 2)$, $a_i \in \{0,1\}^d$ chosen unif. rand **Goal: find *x*** ***Either memory** d² **or** exp(d) **datapoints.***	Same as i.i.d. data setting, but datapoints stored in read-only memory.
Continuous setting Solve Ax=b over **R**^{d} (or regression)	Unknown $x \in \mathrm{R}^d$ with |*x*|=1 chosen uniformly at random. Each datapoint: $(a_i, < a_i, x > +noise)$ $a_i \in \mathrm{R}^d$ chosen e.g. from $N(0, I_d)$ [or some other distribution] Goal: approximate *x*	Same as i.i.d. data setting, but datapoints stored in read-only memory.

	i.i.d. data samples	"Cell-Probe" Model
Discrete setting Solve Ax=b over a finite field	Unknown $x \in \{0,1\}^d$ chosen uniformly at random. Each datapoint: $(a_i, \ < a_i, x > \bmod 2)$, $a_i \in \{0,1\}^d$ chosen unif. rand **Goal: find *x*** **Either memory *d^2 or exp(d)* *datapoints.***	Same as i.i.d. data setting, but datapoints stored in read-only memory.
Continuous setting Solve Ax=b over **R**^{d} (or regression)	Unknown $x \in \mathrm{R}^d$ with |*x*|=1 chosen uniformly at random. Each datapoint: $(a_i, \ < a_i, x > \ + noise)$ $a_i \in \mathrm{R}^d$ chosen e.g. from $N(0, I_d)$ [or some other distribution] Goal: approximate *x*	Same as i.i.d. data setting, but datapoints stored in read-only memory.

	i.i.d. data samples	"Cell-Probe" Model
Discrete setting Solve Ax=b over a finite field	Unknown $x \in \{0,1\}^d$ chosen uniformly at random. Each datapoint: $(a_i, \ < a_i, x > \bmod 2)$, $a_i \in \{0,1\}^d$ chosen unif. rand **Goal: find *x*** ***Either memory d^2 or exp(d) datapoints.***	Same as i.i.d. data setting, but datapoints stored in read-only memory. Easy: linear space and poly(d) queries.
Continuous setting Solve Ax=b over $\mathbf{R}^d$ (or regression)	Unknown $x \in \mathrm{R}^d$ with $\|x\|=1$ chosen uniformly at random. Each datapoint: $(a_i, \ < a_i, x > \ + noise)$ $a_i \in \mathrm{R}^d$ chosen e.g. from $N(0, I_d)$ [or some other distribution] Goal: approximate *x*	Same as i.i.d. data setting, but datapoints stored in read-only memory.

	i.i.d. data samples	"Cell-Probe" Model
Discrete setting Solve Ax=b over a finite field	Unknown $x \in \{0,1\}^d$ chosen uniformly at random. Each datapoint: $(a_i, \ < a_i, x > \bmod 2)$, $a_i \in \{0,1\}^d$ chosen unif. rand **Goal: find *x*** ***Either memory d² or exp(d) datapoints.***	Same as i.i.d. data setting, but datapoints stored in read-only memory. Easy: linear space and poly(d) queries.
Continuous setting Solve Ax=b over $\mathbf{R}^d$ (or regression)	Unknown $x \in \mathrm{R}^d$ with $\lvert x \rvert=1$ chosen uniformly at random. Each datapoint: $(a_i, \ < a_i, x > \ + noise)$ $a_i \in \mathrm{R}^d$ chosen e.g. from $N(0, I_d)$ [or some other distribution] Goal: approximate *x*	Same as i.i.d. data setting, but datapoints stored in read-only memory.

Linear regression: core ML and convex optimization problem

Broader Context:

1^{st} order methods (linear memory, more samples)

vs 2^{nd} order methods (quadratic memory, less samples)

Huge effort to find optimization algorithms with linear memory, that behave like quadratic-memory algorithms (e.g. conjugate gradient…)

Largely unexplored frontier of continuous optimization research:

- Vast literature on lower bounds on #queries required from oracle
- Relatively recent work with *linear* memory: [Dagan-Kur-Shamir'19] on memory lower bounds in streaming model, sparse linear regression [Steinhardt-Duchi'15]

Little on memory/sample tradeoffs for optimization with super-linear memory.

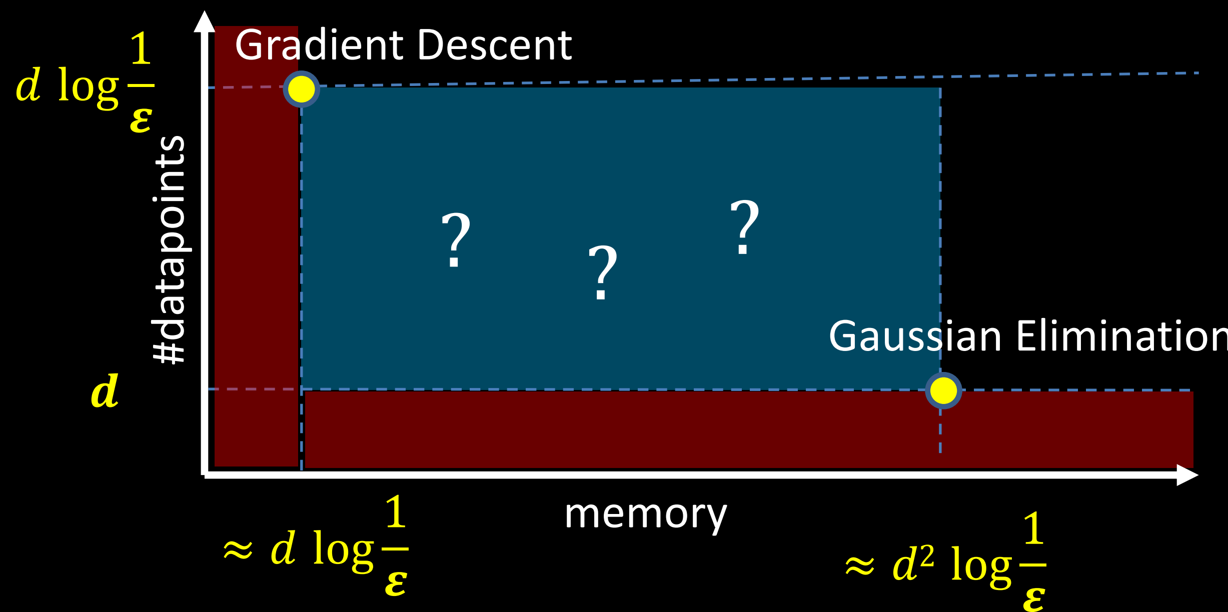
Memory/Data Tradeoffs for Linear Regression

Unknown $x \in \mathbb{R}^d$ with $\| x \| = 1$, chosen uniformly at random.

Given access to stream of examples (a_i, b_i),

$$a_i \sim N(0, I_d)$$

$$b_i = < a_i, x > + \eta_i \qquad \text{noise: } \eta_i \sim Unif[-2^{-d}, 2^{-d}]$$



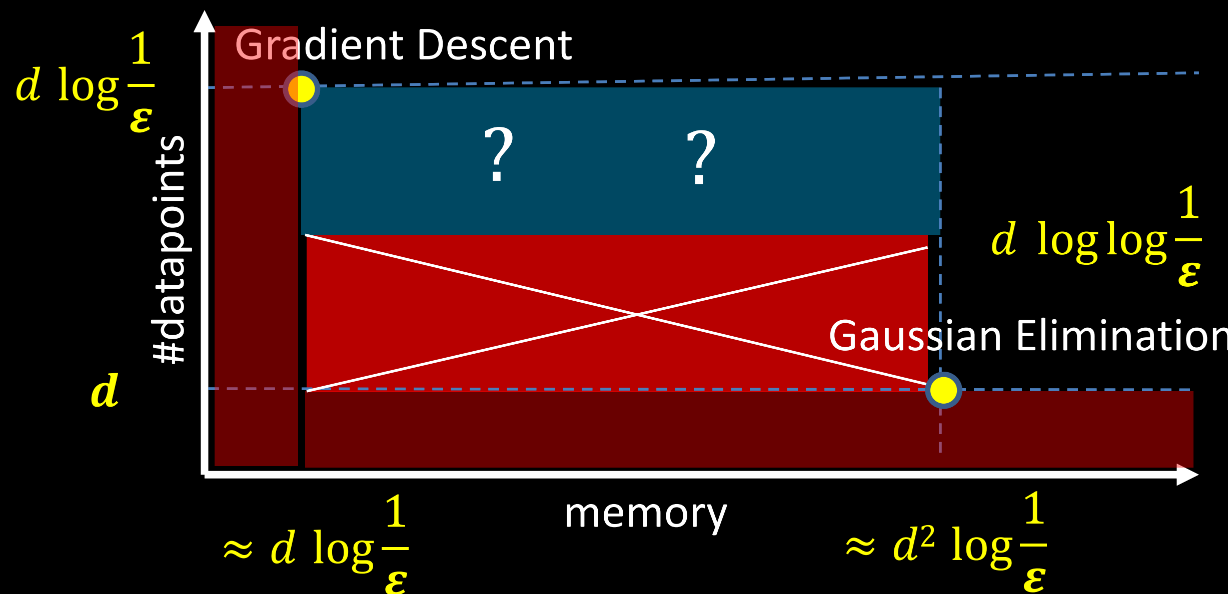
Memory/Data Tradeoffs for Linear Regression

Unknown $x \in \mathbb{R}^d$ with $\| x \| = 1$, chosen uniformly at random.

Given access to stream of examples (a_i, b_i),

$$a_i \sim N(0, I_d)$$

$$b_i = < a_i, x > + \eta_i \qquad \text{noise: } \eta_i \sim Unif[-2^{-d}, 2^{-d}]$$



Theorem [Sharan, Sidford, Valiant '19]:

Any algorithm with $o(d^2)$ memory needs at least $\Omega\left(d \log\log \frac{1}{\varepsilon}\right)$ samples to approximate x with L_2 error ε.

	i.i.d. data samples	"Cell-Probe" Model
Discrete setting Solve Ax=b over a finite field	Unknown $x \in \{0,1\}^d$ chosen uniformly at random. Each datapoint: $(a_i, \ < a_i, x > \bmod 2)$, $a_i \in \{0,1\}^d$ chosen unif. rand **Goal: find *x*** ***Either memory d² or exp(d) datapoints.***	Same as i.i.d. data setting, but datapoints stored in read-only memory. Easy: linear space and poly(d) queries.
Continuous setting Solve Ax=b over $\mathbf{R}^d$ (or regression)	Each datapoint: $(a_i, \ < a_i, x > \ +noise)$ $a_i \in \mathrm{R}^d$ chosen e.g. from $\mathbf{N(0,I_d)}$ **Goal: approximate *x to within* ε** Thm: $o(d^2)$ space implies need d log log $1/\varepsilon$ datapoints (with noise) Conjecture I: $o(d^2)$ space implies need d log $1/\varepsilon$ datapoints Conj. II: $o(d^2)$ space implies need poly(condition number) datapoints	Same as i.i.d. data setting, but datapoints stored in read-only memory.

	i.i.d. data samples	"Cell-Probe" Model
Discrete setting Solve Ax=b over a finite field	Unknown $x \in \{0,1\}^d$ chosen uniformly at random. Each datapoint: $(a_i, < a_i, x > \bmod 2)$, $a_i \in \{0,1\}^d$ chosen unif. rand **Goal: find *x*** ***Either memory d² or exp(d) datapoints.***	Same as i.i.d. data setting, but datapoints stored in read-only memory. Easy: linear space and poly(d) queries.
Continuous setting Solve Ax=b over $\mathbf{R}^d$ (or regression)	Each datapoint: $(a_i, < a_i, x > +noise)$ $a_i \in \mathrm{R}^d$ chosen e.g. from $\mathbf{N(0,I_d)}$ **Goal: approximate *x to within ε*** Thm: $o(d^2)$ space implies need d log log 1/ε datapoints (with noise) Conjecture I: $o(d^2)$ space implies need d log 1/ε datapoints Conj. II: $o(d^2)$ space implies need poly(condition number) datapoints	Same as i.i.d. data setting, but datapoints stored in read-only memory.

	i.i.d. data samples	"Cell-Probe" Model
Discrete setting Solve Ax=b over a finite field	Unknown $x \in \{0,1\}^d$ chosen uniformly at random. Each datapoint: $(a_i, < a_i, x > \bmod 2)$, $a_i \in \{0,1\}^d$ chosen unif. rand **Goal: find *x*** ***Either memory d² or exp(d) datapoints.***	Same as i.i.d. data setting, but datapoints stored in read-only memory. Easy: linear space and poly(d) queries.
Continuous setting Solve Ax=b over **R**^{d} (or regression)	Each datapoint: $(a_i, < a_i, x > +noise)$ $a_i \in \mathrm{R}^d$ chosen e.g. from **N(0,I$_d$)** **Goal: approximate *x to within* ε** Thm: o(d^2) space implies need d log log 1/ε datapoints (with noise) Conjecture I: o(d^2) space implies need d log 1/ε datapoints Conj. II: o(d^2) space implies need poly(condition number) datapoints	Same as i.i.d. data setting, but datapoints stored in read-only memory. Conjecture: Need super-linear space. [Might be *very* hard to prove!!!!] Approach of the discrete setting fails because numerical errors grow without poly(d) precision arithmetic