Volume sampling for linear regression

Michał Dereziński
Manfred Warmuth

UC Santa Cruz
June 12, 2018

Why linear regression?

It's quadratic - so all should be known!

- New sampling routines (variants of volume sampling)
- New fundamental matrix identities
- Unbiased estimators based on small samples
- Scale free multiplicative loss bounds
- New efficient algorithms
- Key implicit motivation: Newton's method

Why linear regression?

It's quadratic - so all should be known!

- New sampling routines (variants of volume sampling)
- New fundamental matrix identities
- Unbiased estimators based on small samples
- Scale free multiplicative loss bounds
- New efficient algorithms
- Key implicit motivation: Newton's method

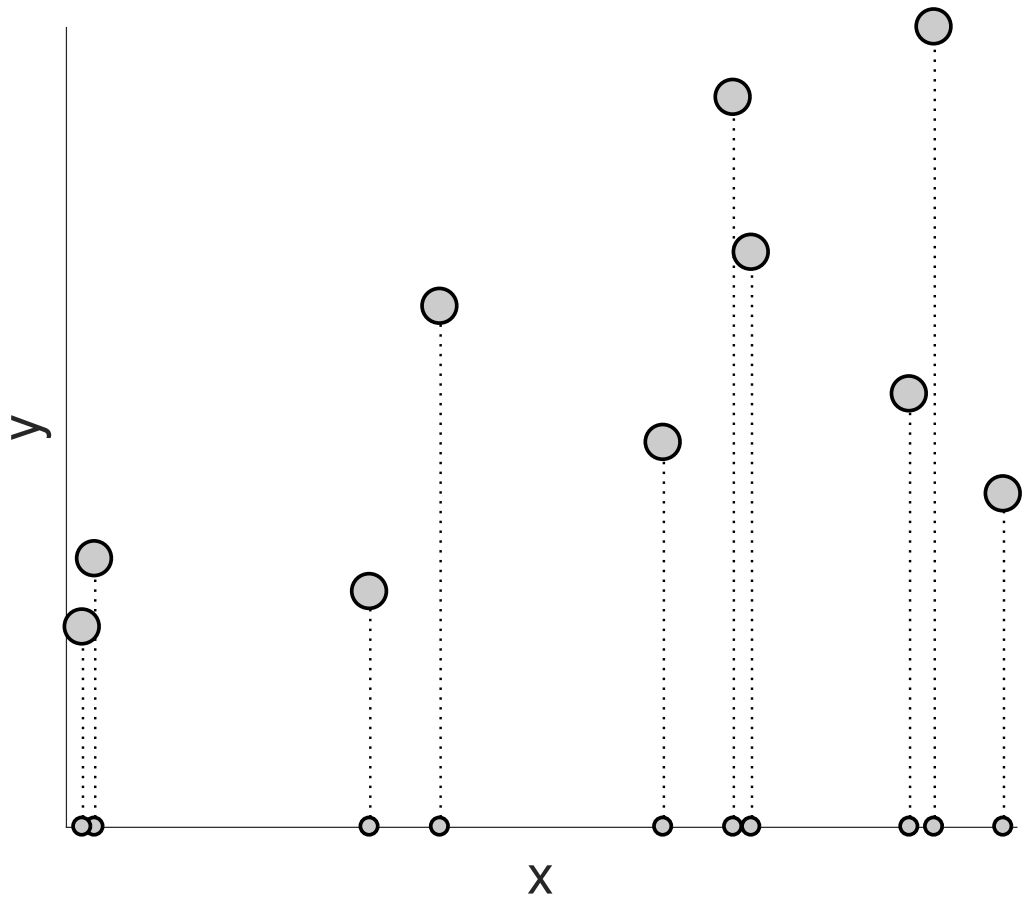
Why linear regression?

It's quadratic - so all should be known!

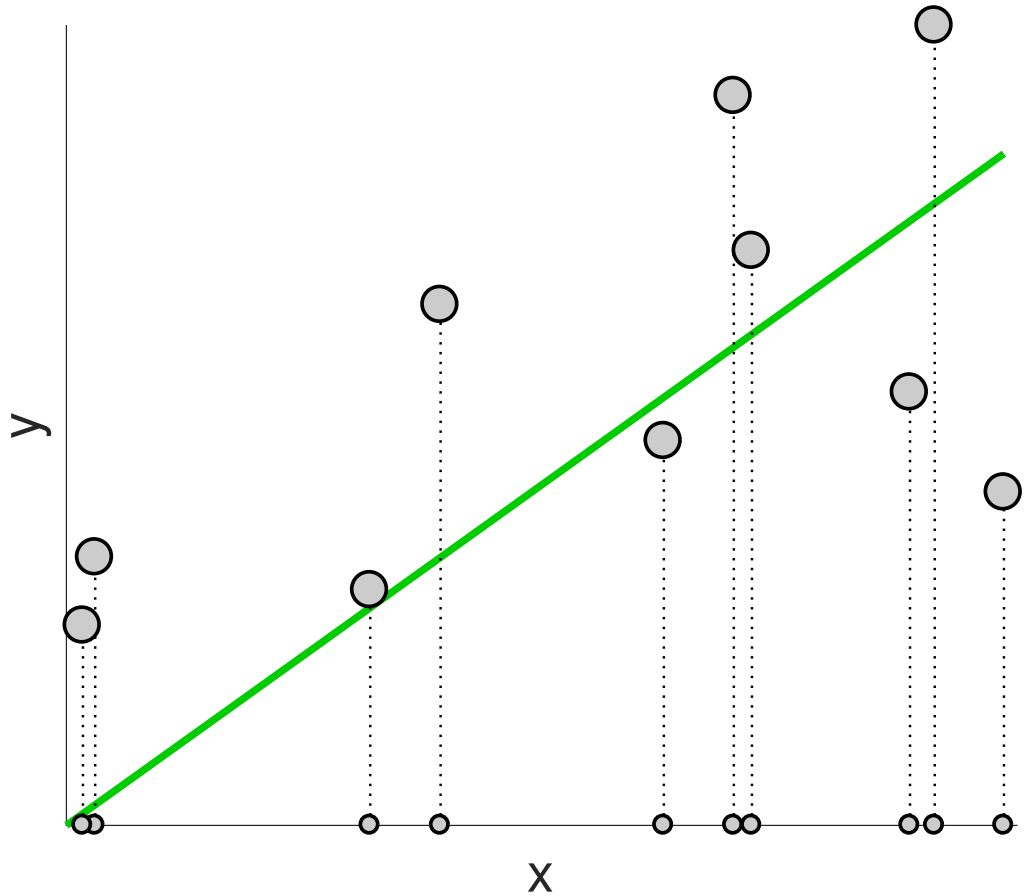
- New sampling routines (variants of volume sampling)
- New fundamental matrix identities
- Unbiased estimators based on small samples
- Scale free multiplicative loss bounds
- New efficient algorithms
- Key implicit motivation: Newton's method

Outline

Linear regression

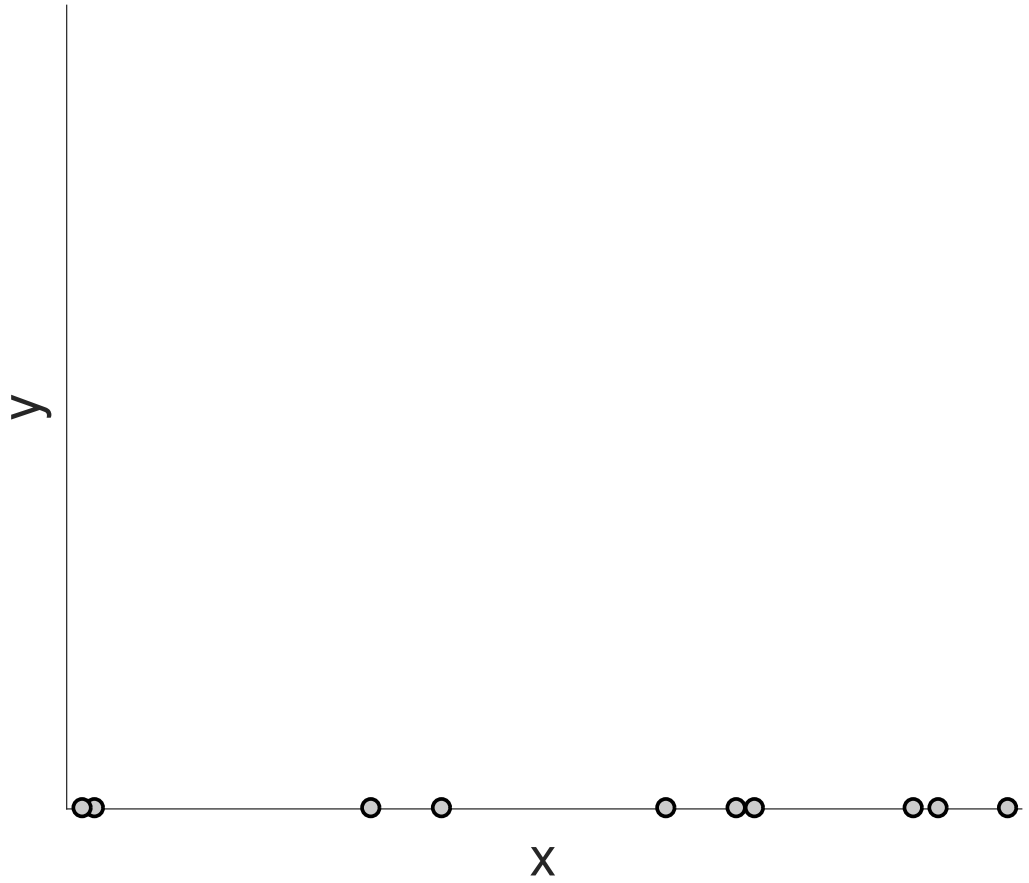


Optimal solution



$$w^* = \underset{w}{\operatorname{argmin}} \sum_i (x_i w - y_i)^2$$

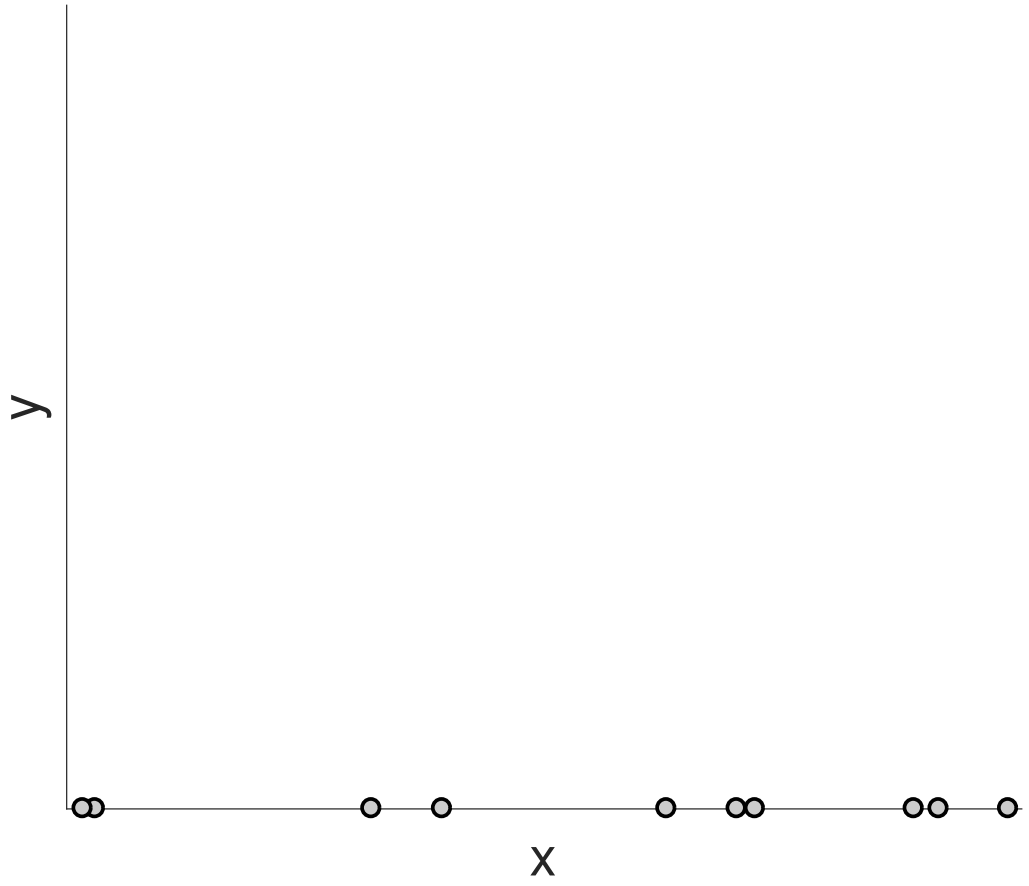
How many labels needed to get close to optimum?



- All x_i given
- But labels y_i unknown

Guess how many needed?

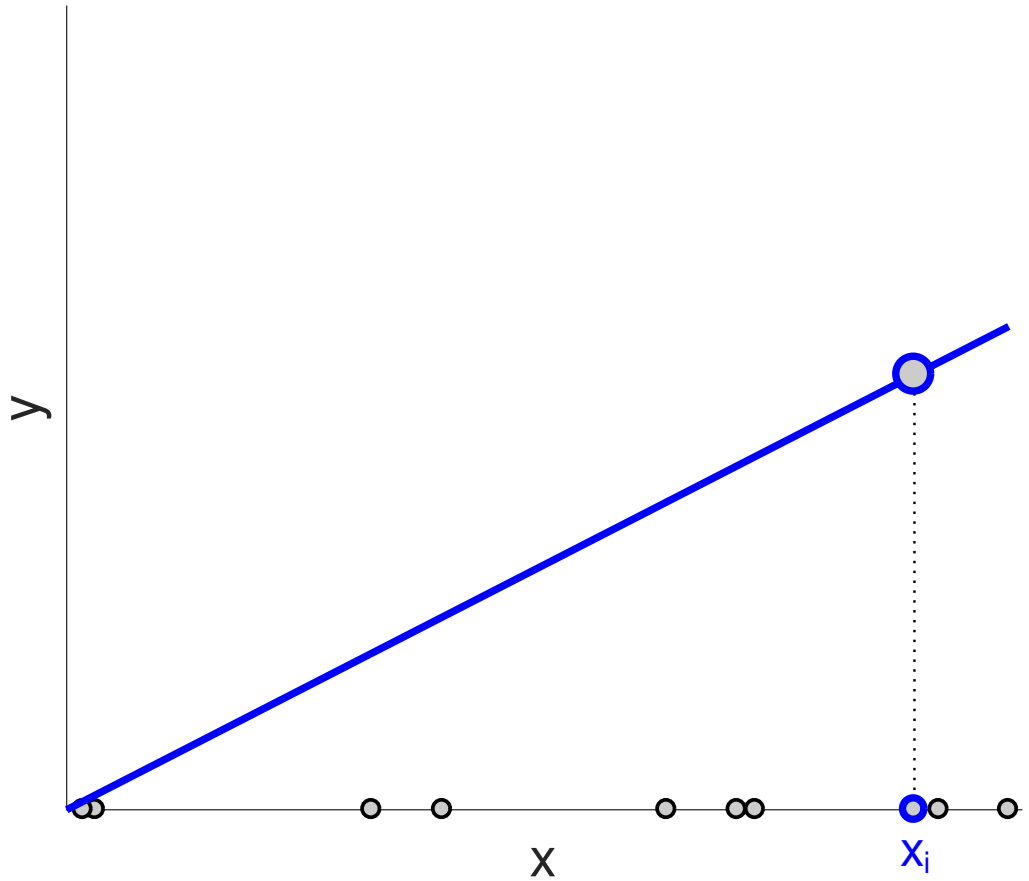
How many labels needed to get close to optimum?



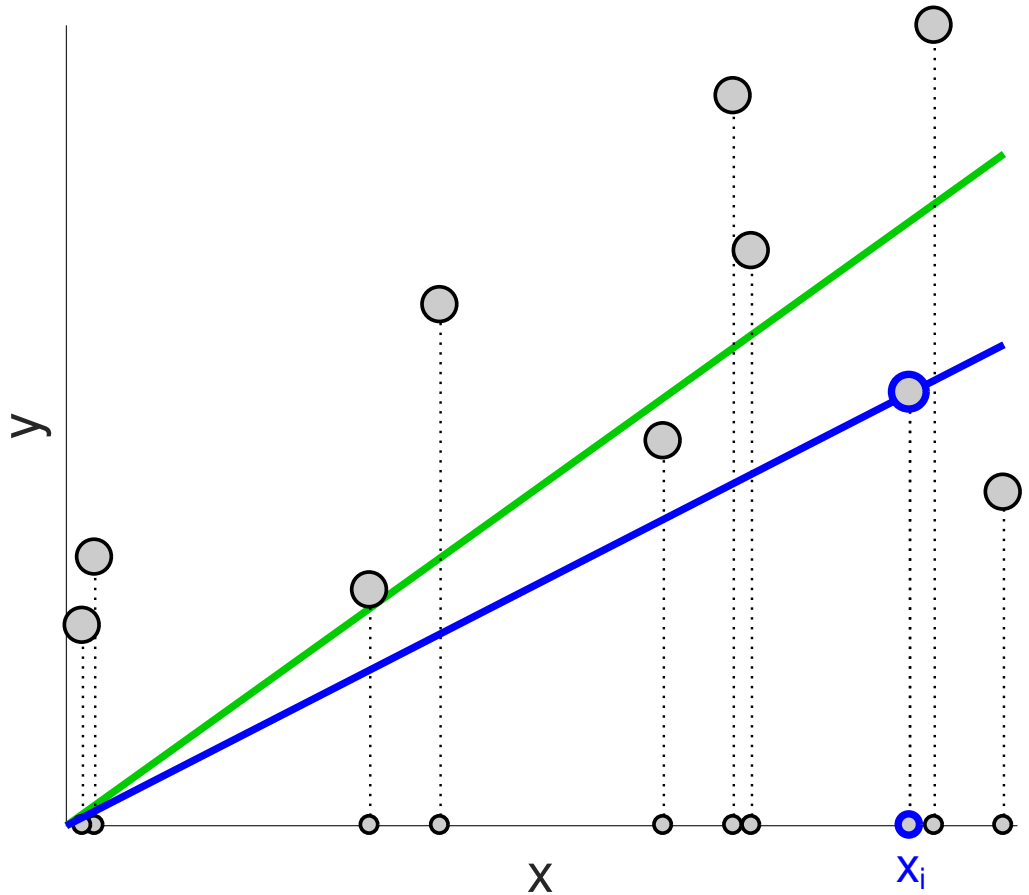
- All x_i given
- But labels y_i unknown

Guess how many needed?

Answer: one label

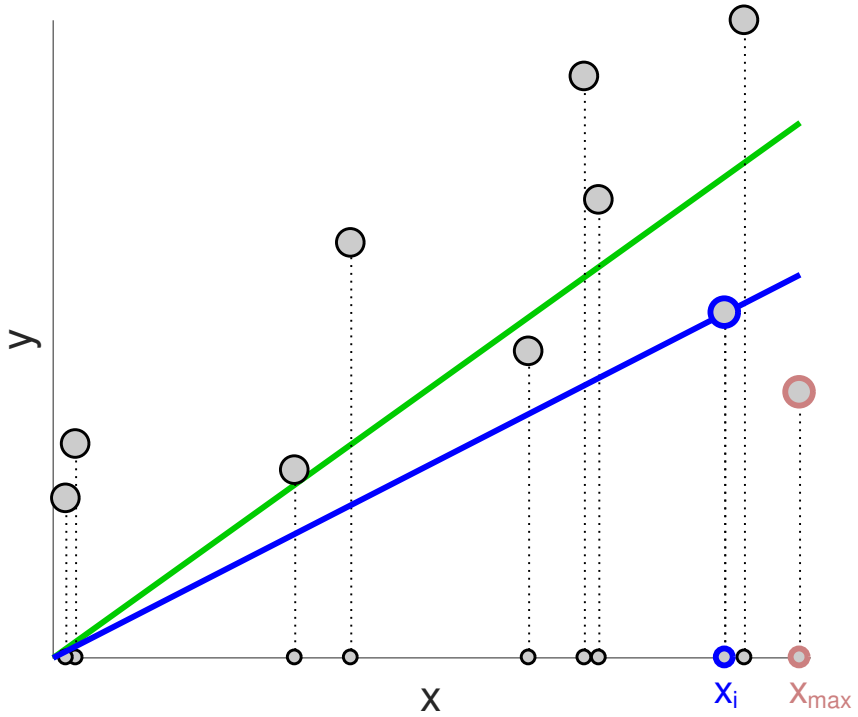


How good is one label?



Loss of estimate $= 2 \times$ Loss of optimum

Which one?



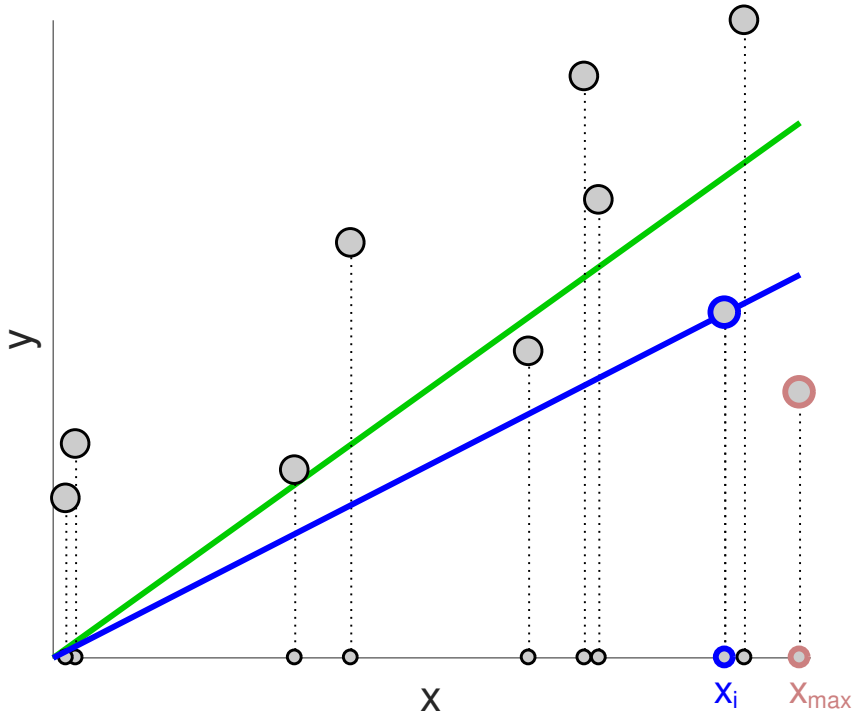
- x_{max} (furthest from 0) is bad
- any deterministic choice is bad

Good: 1 label y_i drawn $\sim x_i^2$

$$\mathbb{E}_i \sum_j \Big(\underbrace{\frac{y_i}{x_i}}_{w_i^*} x_j - y_j \Big)^2 = 2 \sum_j (w^* x_j - y_j)^2$$

$$\mathbb{E}_i \, w_i^* = \sum_i \overbrace{\frac{x_i^2}{\|\mathbf{x}\|^2}}^{P(i)} \overbrace{\frac{y_i}{x_i}}^{w_i^*} = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|^2} = w^*$$

Which one?



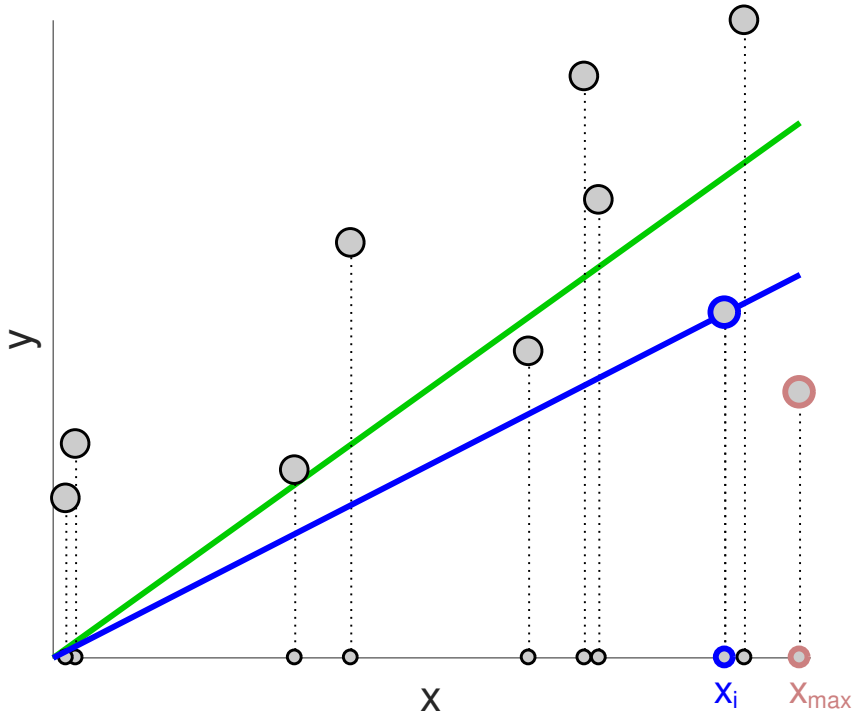
- x_{max} (furthest from 0) is bad
- any deterministic choice is bad

Good: 1 label y_i drawn $\sim x_i^2$

$$\mathbb{E}_i \sum_j \Big(\underbrace{\frac{y_i}{x_i}}_{w_i^*} x_j - y_j \Big)^2 = 2 \sum_j (w^* x_j - y_j)^2$$

$$\mathbb{E}_i \, w_i^* = \sum_i \overbrace{\frac{x_i^2}{\|\mathbf{x}\|^2}}^{P(i)} \overbrace{\frac{y_i}{x_i}}^{w_i^*} = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|^2} = w^*$$

Which one?



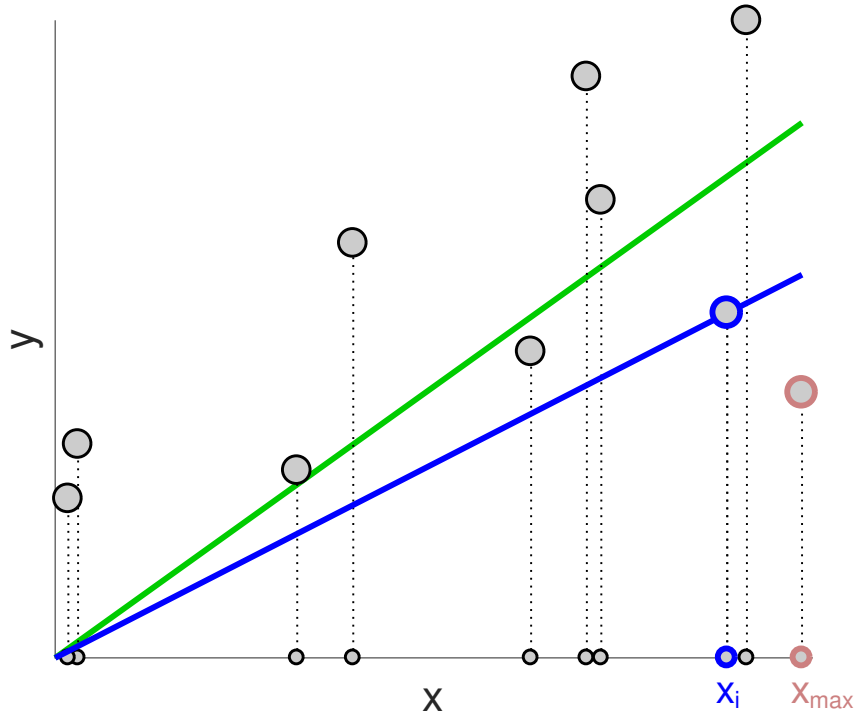
- x_{max} (furthest from 0) is bad
- any deterministic choice is bad

Good: 1 label y_i drawn $\sim x_i^2$

$$\mathbb{E}_i \sum_j \Big(\underbrace{\frac{y_i}{x_i}}_{w_i^*} x_j - y_j \Big)^2 = 2 \sum_j (w^* x_j - y_j)^2$$

$$\mathbb{E}_i \, w_i^* = \sum_i \overbrace{\frac{x_i^2}{\|\mathbf{x}\|^2}}^{P(i)} \overbrace{\frac{y_i}{x_i}}^{w_i^*} = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|^2} = w^*$$

Which one?



- x_{max} (furthest from 0) is bad
- any deterministic choice is bad

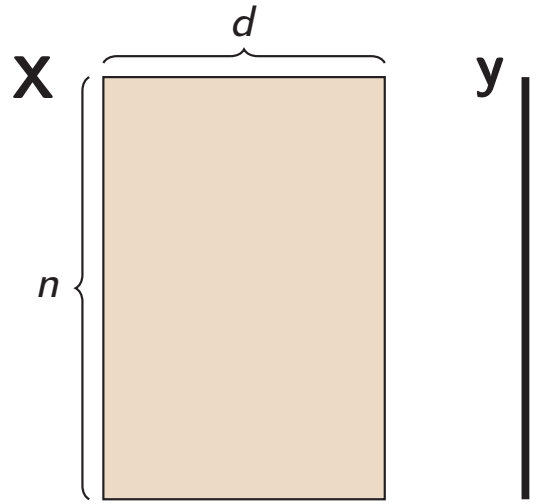
Good: 1 label y_i drawn $\sim x_i^2$

$$\mathbb{E}_i \sum_j \Big(\underbrace{\frac{y_i}{x_i}}_{w_i^*} x_j - y_j \Big)^2 = 2 \sum_j (w^* x_j - y_j)^2$$

$$\mathbb{E}_i \, w_i^* = \sum_i \overbrace{\frac{x_i^2}{\|\mathbf{x}\|^2}}^{P(i)} \overbrace{\frac{y_i}{x_i}}^{w_i^*} = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|^2} = w^*$$

General: sub-sampling for linear regression

Given: n points $\mathbf{x}_i \in \mathbb{R}^d$ with hidden labels $y_i \in \mathbb{R}$
Goal: Minimize loss $L(\mathbf{w}) = \sum_i (\mathbf{x}_i^\top \mathbf{w} - y_i)^2$ over all n points



Strategy: Solve subproblem $(\mathbf{X}_S, \mathbf{y}_S)$, obtaining:

$$\mathbf{w}^*(S) = \operatorname*{argmin}_{\mathbf{w}} \sum_{i \in S} (\mathbf{x}_i^\top \mathbf{w} - y_i)^2 = (\mathbf{X}_S)^+ \mathbf{y}_S$$

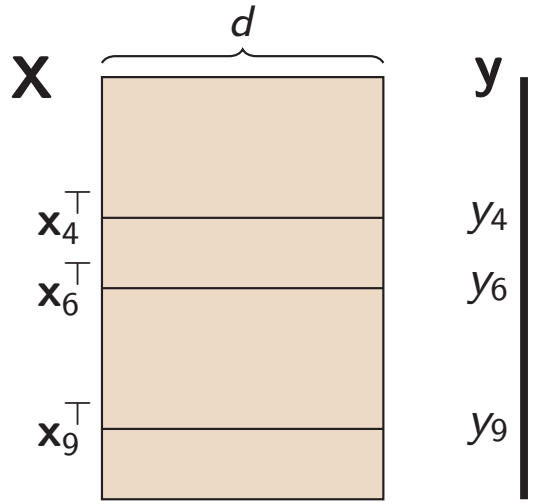
$(\mathbf{X}_S)^+ = (\mathbf{X}_S^\top \mathbf{X}_S)^{-1} \mathbf{X}_S^\top$ - pseudo-inverse of $\mathbf{X}_S$

General: sub-sampling for linear regression

Given: n points $\mathbf{x}_i \in \mathbb{R}^d$ with hidden labels $y_i \in \mathbb{R}$
Goal: Minimize loss $L(\mathbf{w}) = \sum_i (\mathbf{x}_i^\top \mathbf{w} - y_i)^2$ over all n points

Sample
$S = \{4, 6, 9\}$

Receive y_4, y_6, y_9



Strategy: Solve subproblem $(\mathbf{X}_S, \mathbf{y}_S)$, obtaining:

$$\mathbf{w}^*(S) = \operatorname*{argmin}_{\mathbf{w}} \sum_{i \in S} (\mathbf{x}_i^\top \mathbf{w} - y_i)^2 = (\mathbf{X}_S)^+ \mathbf{y}_S$$

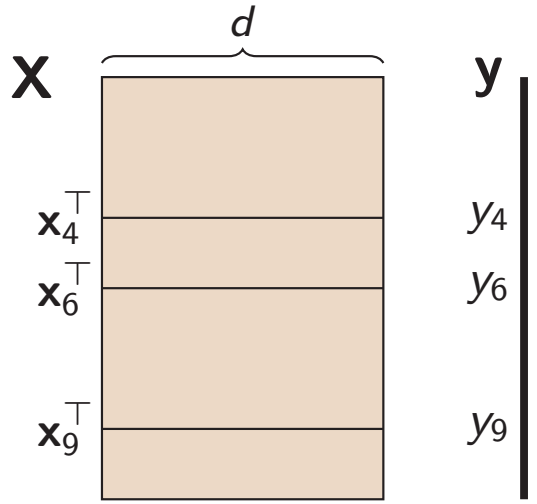
$(\mathbf{X}_S)^+ = (\mathbf{X}_S^\top \mathbf{X}_S)^{-1} \mathbf{X}_S^\top$ - pseudo-inverse of $\mathbf{X}_S$

General: sub-sampling for linear regression

Given: n points $\mathbf{x}_i \in \mathbb{R}^d$ with hidden labels $y_i \in \mathbb{R}$
Goal: Minimize loss $L(\mathbf{w}) = \sum_i (\mathbf{x}_i^\top \mathbf{w} - y_i)^2$ over all n points

Sample
$S = \{4, 6, 9\}$

Receive y_4, y_6, y_9



Strategy: Solve subproblem $(\mathbf{X}_S, \mathbf{y}_S)$, obtaining:

$$\mathbf{w}^*(S) = \operatorname*{argmin}_{\mathbf{w}} \sum_{i \in S} (\mathbf{x}_i^\top \mathbf{w} - y_i)^2 = (\mathbf{X}_S)^+ \mathbf{y}_S$$

$$(\mathbf{X}_S)^+ = (\mathbf{X}_S^\top \mathbf{X}_S)^{-1} \mathbf{X}_S^\top \quad \text{- pseudo-inverse of } \mathbf{X}_S$$

Need of randomness

1-dimensional example:

$$\mathbf{X}^\top = \begin{pmatrix} \overset{\mathbf{x}_1}{1} & \overset{\mathbf{x}_2}{1} & \cdots & \overset{\mathbf{x}_n}{1} \end{pmatrix}$$
$$\mathbf{y}^\top = \begin{pmatrix} 0 & 1 & \cdots & 1 \end{pmatrix}$$

Deterministic pick $S = \{1\}$, receive $y_1 = 0$
Deterministic predictor $\mathbf{w}^*(\{1\}) = 0$
Optimal predictor $\mathbf{w}^* = \frac{n-1}{n} = 1 - \frac{1}{n}$

$$\underbrace{L(\overbrace{\mathbf{w}^*(\{1\})}^{0})}_{n-1} = n \underbrace{L(\overbrace{\mathbf{w}^*}^{\frac{n-1}{n}})}_{\frac{n-1}{n}}$$

Thus: No good deterministic algorithm for sampling d labels.

With uniform choice of $S : |S| = 1$, $\mathbb{E}[L(\mathbf{w}^*(S))] = 2L(\mathbf{w}^*)$

Need of randomness

1-dimensional example:

$$\mathbf{X}^\top = \begin{pmatrix} \overset{\mathbf{x}_1}{1} & \overset{\mathbf{x}_2}{1} & \cdots & \overset{\mathbf{x}_n}{1} \end{pmatrix}$$

$$\mathbf{y}^\top = \begin{pmatrix} 0 & 1 & \cdots & 1 \end{pmatrix}$$

Deterministic pick $S = \{1\}$, receive $y_1 = 0$
Deterministic predictor $\mathbf{w}^*(\{1\}) = 0$
Optimal predictor $\mathbf{w}^* = \frac{n-1}{n} = 1 - \frac{1}{n}$

$$\underbrace{L(\overbrace{\mathbf{w}^*(\{1\})}^{0})}_{n-1} = n \underbrace{L(\overbrace{\mathbf{w}^*}^{\frac{n-1}{n}})}_{\frac{n-1}{n}}$$

Thus: No good deterministic algorithm for sampling d labels.

With uniform choice of $S : |S| = 1$, $\mathbb{E}[L(\mathbf{w}^*(S))] = 2L(\mathbf{w}^*)$

Need of randomness

1-dimensional example:

$$\mathbf{X}^\top = \begin{pmatrix} \overset{\mathbf{x}_1}{1} & \overset{\mathbf{x}_2}{1} & \cdots & \overset{\mathbf{x}_n}{1} \end{pmatrix}$$

$$\mathbf{y}^\top = \begin{pmatrix} 0 & 1 & \cdots & 1 \end{pmatrix}$$

Deterministic pick $S = \{1\}$, receive $y_1 = 0$
Deterministic predictor $\mathbf{w}^*(\{1\}) = 0$
Optimal predictor $\mathbf{w}^* = \frac{n-1}{n} = 1 - \frac{1}{n}$

$$\underbrace{L(\overbrace{\mathbf{w}^*(\{1\})}^{0})}_{n-1} = n \underbrace{L(\overbrace{\mathbf{w}^*}^{\frac{n-1}{n}})}_{\frac{n-1}{n}}$$

Thus: No good deterministic algorithm for sampling d labels.

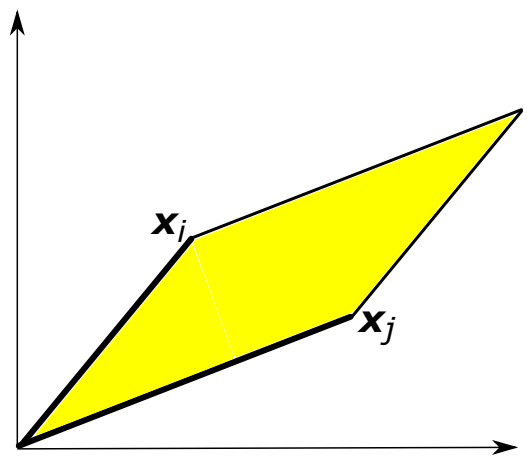
With uniform choice of $S : |S| = 1$, $\mathbb{E}[L(\mathbf{w}^*(S))] = 2L(\mathbf{w}^*)$

Volume Sampling [DRVW06]

For $S = \{i, j\}$:

$$\mathbf{X}_S = \begin{pmatrix} - & \mathbf{x}_i^\top & - \\ - & \mathbf{x}_j^\top & - \end{pmatrix}$$

$\det(\mathbf{X}_S^\top \mathbf{X}_S) =$ **squared** volume spanned by $\{\mathbf{x}_i, \mathbf{x}_j\}$



Distribution over all d-element subsets S:

$$P(S) = \det(\mathbf{X}_S^\top \mathbf{X}_S) \,/\, Z$$

Note: Normalization factor Z can be derived using Cauchy-Binet formula:

$$Z = \sum_{S:|S|=d} \det(\mathbf{X}_S^\top \mathbf{X}_S) = \det(\mathbf{X}^\top \mathbf{X})$$

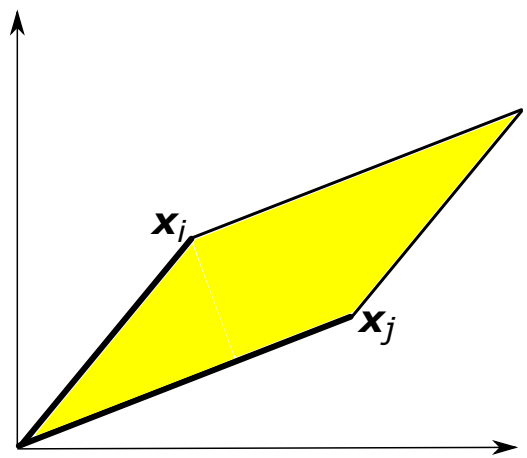
Generalizes to larger set sizes [AB13]

Volume Sampling [DRVW06]

For $S = \{i, j\}$:

$$\mathbf{X}_S = \begin{pmatrix} - & \mathbf{x}_i^\top & - \\ - & \mathbf{x}_j^\top & - \end{pmatrix}$$

$\det(\mathbf{X}_S^\top \mathbf{X}_S) =$ **squared** volume spanned by $\{\mathbf{x}_i, \mathbf{x}_j\}$



Distribution over all d-element subsets S:

$$P(S) = \det(\mathbf{X}_S^\top \mathbf{X}_S) \,/\, Z$$

Note: Normalization factor Z can be derived using Cauchy-Binet formula:

$$Z = \sum_{S:|S|=d} \det(\mathbf{X}_S^\top \mathbf{X}_S) = \det(\mathbf{X}^\top \mathbf{X})$$

Generalizes to larger set sizes [AB13]

Volume Sampling [DRVW06]

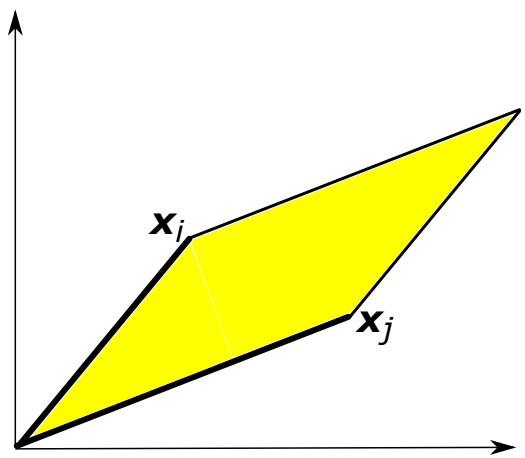
For $S = \{i, j\}$:

$$\mathbf{X}_S = \begin{pmatrix} - & \mathbf{x}_i^\top & - \\ - & \mathbf{x}_j^\top & - \end{pmatrix}$$

$\det(\mathbf{X}_S^\top \mathbf{X}_S) =$ **squared** volume spanned by $\{\mathbf{x}_i, \mathbf{x}_j\}$

Distribution over all d-element subsets S:

$$P(S) = \det(\mathbf{X}_S^\top \mathbf{X}_S) \,/\, Z$$



Note: Normalization factor Z can be derived using Cauchy-Binet formula:

$$Z = \sum_{S:|S|=d} \det(\mathbf{X}_S^\top \mathbf{X}_S) = \det(\mathbf{X}^\top \mathbf{X})$$

Generalizes to larger set sizes [AB13]

Outline

Linear regression with dimension many labels

Theorem ([DW17])

For a volume-sampled d-element set S,

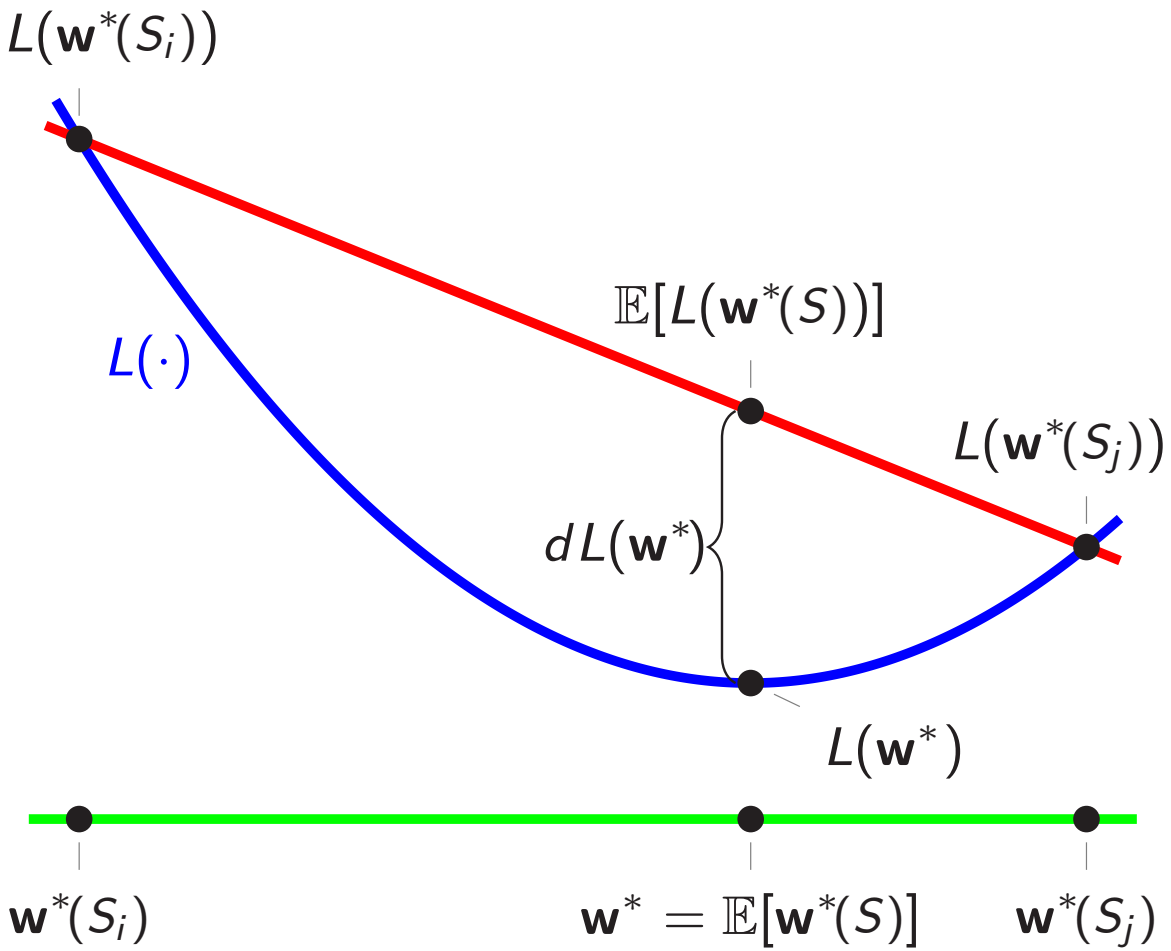
$$\mathbb{E}\left[L(\mathbf{w}^*(S))\right] = (d+1)\, L(\underbrace{\mathbf{w}^*}_{\mathbb{E}[\mathbf{w}^*(S)]}),$$

if $\mathbf{X}$ is in general position

- Sampling distribution does not depend on the labels
- No range restrictions! **No dependence on** n

A gap in Jensen's inequality (when $\mathbf{X}$ in general position)

$$\underbrace{\mathbb{E}[L(\mathbf{w}^*(S))] - L(\overbrace{\mathbf{w}^*}^{\mathbb{E}[\mathbf{w}^*(S)]})}_{\text{regret}} = \underbrace{d\,L(\mathbf{w}^*)}_{\text{gap in Jensen's}} = \underbrace{\mathbb{E}[\|\mathbf{X}\mathbf{w}^*(S) - \mathbf{X}\mathbf{w}^*\|^2]}_{\text{variance}}$$



What about iid sampling?

Main candidate: iid sampling using leverage scores [DMM06]
Widely used for linear regression

$$\text{leverage score of } \mathbf{x}_i \quad \propto \quad \mathbf{x}_i^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_i$$

Leverage scores are marginals of size d volume sampling

Problems with iid sampling:

1. requires at least $d \log d$ labels
2. produces biased estimators

Volume sampling

Requires only d labels and produces unbiased estimators,
with runtime comparable to leverage scores

What about iid sampling?

Main candidate: iid sampling using leverage scores [DMM06]
Widely used for linear regression

$$\text{leverage score of } \mathbf{x}_i \quad \propto \quad \mathbf{x}_i^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_i$$

Leverage scores are marginals of size d volume sampling

Problems with iid sampling:

1. requires at least $d \log d$ labels
2. produces biased estimators

Volume sampling

Requires only d labels and produces unbiased estimators,
with runtime comparable to leverage scores

What about iid sampling?

Main candidate: iid sampling using leverage scores [DMM06]
Widely used for linear regression

$$\text{leverage score of } \mathbf{x}_i \quad \propto \quad \mathbf{x}_i^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_i$$

Leverage scores are marginals of size d volume sampling

Problems with iid sampling:

1. requires at least $d \log d$ labels
2. produces biased estimators

Volume sampling

Requires only d labels and produces unbiased estimators,
with runtime comparable to leverage scores

What about iid sampling?

Main candidate: iid sampling using leverage scores [DMM06]
Widely used for linear regression

$$\text{leverage score of } \mathbf{x}_i \quad \propto \quad \mathbf{x}_i^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_i$$

Leverage scores are marginals of size d volume sampling

Problems with iid sampling:

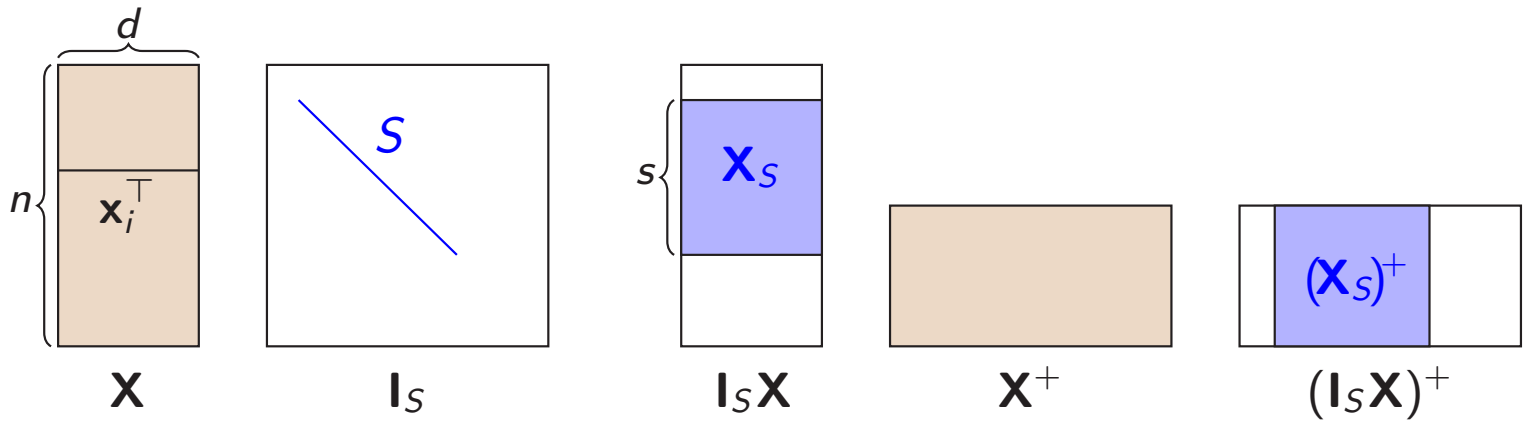
1. requires at least $d \log d$ labels
2. produces biased estimators

Volume sampling

Requires only d labels and produces unbiased estimators,

with runtime comparable to leverage scores

Volume Sampling gives unbiased estimators



Expected pseudo-inverse

$$\mathbb{E}[(\mathbf{I}_S\mathbf{X})^+] = \mathbf{X}^+$$

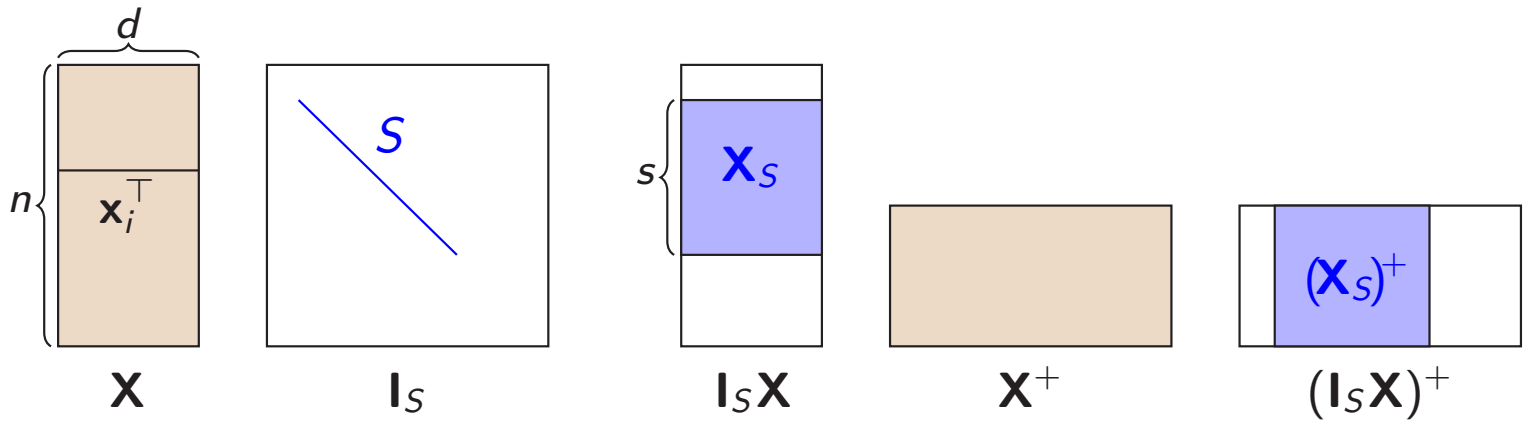
Variance of pseudo-inverse estimator:

$$\mathbb{E}[\underbrace{(\mathbf{I}_S\mathbf{X})^+(\mathbf{I}_S\mathbf{X})^{+\top}}_{(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}}] - \mathbf{X}^+\mathbf{X}^{+\top} = (n-d)\mathbf{X}^+\mathbf{X}^{+\top}$$

$\mathbf{w}^*(S)$ - **unbiased estimator** of $\mathbf{w}^*$

$$\mathbb{E}[\mathbf{w}^*(S)] = \mathbb{E}[(\mathbf{I}_S\mathbf{X})^+]\,\mathbf{y} = \mathbf{X}^+\mathbf{y} = \mathbf{w}^*$$

Volume Sampling gives unbiased estimators



Expected pseudo-inverse

$$\mathbb{E}[(\mathbf{I}_S\mathbf{X})^+] = \mathbf{X}^+$$

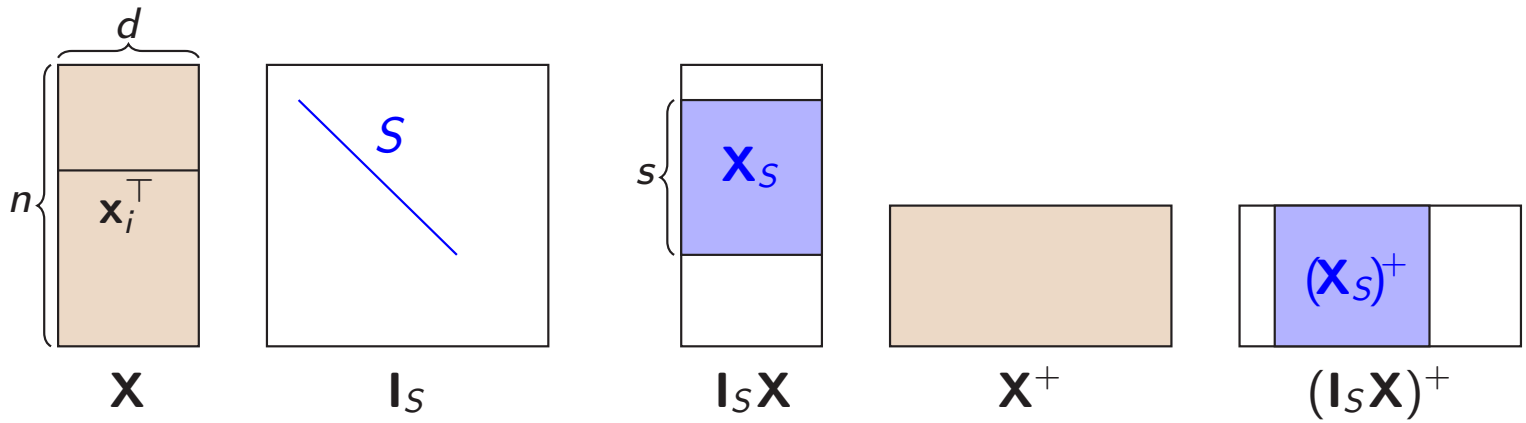
Variance of pseudo-inverse estimator:

$$\mathbb{E}[\underbrace{(\mathbf{I}_S\mathbf{X})^+(\mathbf{I}_S\mathbf{X})^{+\top}}_{(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}}] - \mathbf{X}^+\mathbf{X}^{+\top} = (n-d)\mathbf{X}^+\mathbf{X}^{+\top}$$

$\mathbf{w}^*(S)$ - **unbiased estimator** of $\mathbf{w}^*$

$$\mathbb{E}[\mathbf{w}^*(S)] = \mathbb{E}[(\mathbf{I}_S\mathbf{X})^+]\,\mathbf{y} = \mathbf{X}^+\mathbf{y} = \mathbf{w}^*$$

Volume Sampling gives unbiased estimators



Expected pseudo-inverse

$$\mathbb{E}[(\mathbf{I}_S\mathbf{X})^+] = \mathbf{X}^+$$

Variance of pseudo-inverse estimator:

$$\mathbb{E}[\underbrace{(\mathbf{I}_S\mathbf{X})^+(\mathbf{I}_S\mathbf{X})^{+\top}}_{(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}}] - \mathbf{X}^+\mathbf{X}^{+\top} = (n-d)\mathbf{X}^+\mathbf{X}^{+\top}$$

$\mathbf{w}^*(S)$ - **unbiased estimator** of $\mathbf{w}^*$

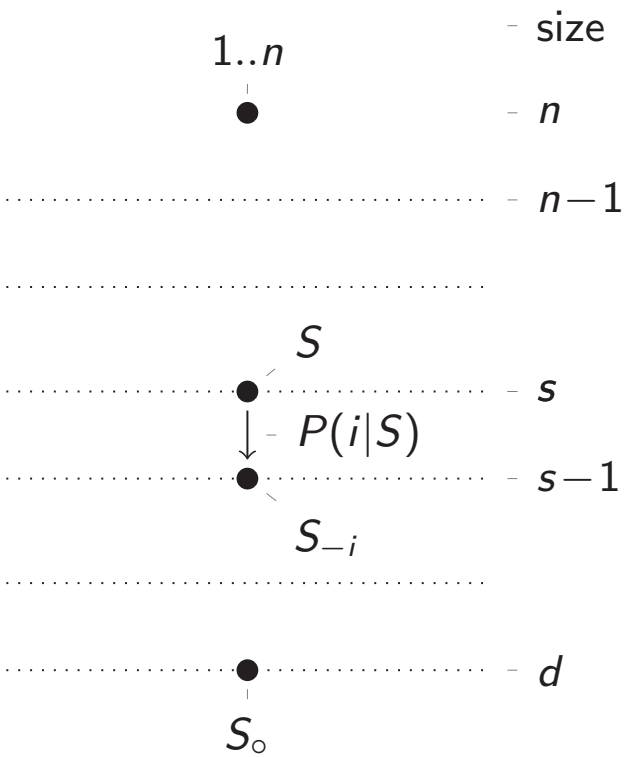
$$\mathbb{E}[\mathbf{w}^*(S)] = \mathbb{E}[(\mathbf{I}_S\mathbf{X})^+]\,\mathbf{y} = \mathbf{X}^+\mathbf{y} = \mathbf{w}^*$$

Implications of the new matrix equalities

- Experimental design (A-optimality condition)
- Matrix approximation (pseudo-inverse estimation)
- New statistical guarantees for linear regression
- Faster algorithms for sampling diverse subsets
- ...

Outline

Reverse Iterative Sampling



Start with $S = \{1..n\}$

Sample index $i \in S$

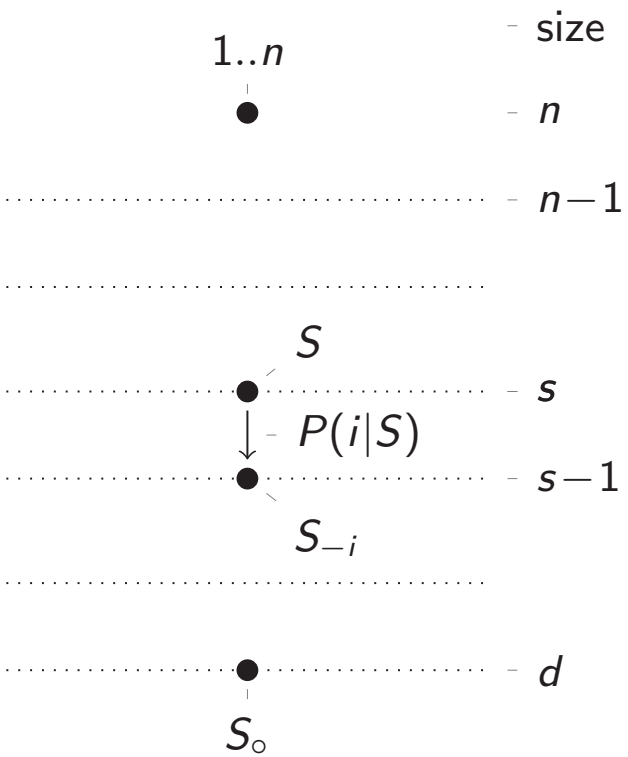
Go to set $S_{-i} = S - \{i\}$

Repeat until desired size

$$P(\{1..n\} \downarrow \downarrow S_\circ) = \prod_{\text{edges of path}} P(i|S)$$

$$P(S_\circ) = \sum_{\text{paths } \{1..n\} \downarrow \downarrow S_\circ} P(\text{path})$$

Reverse Iterative Sampling



Start with $S = \{1..n\}$

Sample index $i \in S$

Go to set $S_{-i} = S - \{i\}$

Repeat until desired size

$$P(\ \overset{\{1..n\}}{\underset{S_\circ}{\downarrow}}\ ) = \prod_{\text{edges of path}} P(i|S)$$

$$P(S_\circ) = \sum_{\text{paths}\ \overset{\{1..n\}}{\underset{S_\circ}{\downarrow}}} P(\text{path})$$

Reverse Iterative Volume Sampling

Thm: Give s edges out of node S the following probabilities:

$$P(i|S) := \frac{\det(\mathbf{X}_{S_{-i}}^\top \mathbf{X}_{S_{-i}})}{(|S| - d)\det(\mathbf{X}_S^\top \mathbf{X}_S)}$$

(volume sampling set one smaller)

($\mathbf{X}_S$ is the sub-matrix of columns indexed by S)

Then the total probability of all paths into $S_\circ$ is

$$P(S_\circ) = \frac{\det(\mathbf{X}_{S_\circ}^\top \mathbf{X}_{S_\circ})}{\binom{n-d}{s-d}\det(\mathbf{X}^\top \mathbf{X})} \qquad \textbf{(volume sampling size } s \textbf{ set)}$$

Proof:

$$P(\overset{\{1..n\}}{\underset{S_\circ}{\downarrow}}) = \overbrace{\prod_{\text{edges of path}} P(i|S)}^{\text{telescoping product}} = \frac{1}{(n-d)\ldots(s-d+1)} \frac{\det(\mathbf{X}_{S_\circ}^\top \mathbf{X}_{S_\circ})}{\det(\mathbf{X}^\top \mathbf{X})}$$

$$P(S_\circ) = \frac{\overbrace{(n-s)!}^{\#\text{ of paths}}}{(n-d)\ldots(s-d+1)} \frac{\det(\mathbf{X}_{S_\circ}^\top \mathbf{X}_{S_\circ})}{\det(\mathbf{X}^\top \mathbf{X})} = \frac{1}{\binom{n-d}{s-d}} \frac{\det(\mathbf{X}_{S_\circ}^\top \mathbf{X}_{S_\circ})}{\det(\mathbf{X}^\top \mathbf{X})}$$ ■

When $s = d$, then this is a short proof for Cauchy Binet:

$$\det(\mathbf{X}^\top \mathbf{X}) = \sum_{S:|S|=d} \det(\mathbf{X}_S^\top \mathbf{X}_S)$$

Proof:

$$P(\begin{array}{c}\{1..n\}\\ \downarrow \\ \downarrow \\ S_\circ\end{array}) = \overbrace{\prod_{\text{edges of path}} P(i|S)}^{\text{telescoping product}} = \frac{1}{(n-d)\ldots(s-d+1)} \frac{\det(\mathbf{X}_{S_\circ}^\top \mathbf{X}_{S_\circ})}{\det(\mathbf{X}^\top \mathbf{X})}$$

$$P(S_\circ) = \frac{\overbrace{(n-s)!}^{\#\text{ of paths}}}{(n-d)\ldots(s-d+1)} \frac{\det(\mathbf{X}_{S_\circ}^\top \mathbf{X}_{S_\circ})}{\det(\mathbf{X}^\top \mathbf{X})} = \frac{1}{\binom{n-d}{s-d}} \frac{\det(\mathbf{X}_{S_\circ}^\top \mathbf{X}_{S_\circ})}{\det(\mathbf{X}^\top \mathbf{X})} \quad \blacksquare$$

When $s = d$, then this is a short proof for Cauchy Binet:

$$\det(\mathbf{X}^\top \mathbf{X}) = \sum_{S:|S|=d} \det(\mathbf{X}_S^\top \mathbf{X}_S)$$

Composition property

Volume Sampling is closed under subsampling

Hierarchical sampling ($t \geq s$):

$$\begin{array}{rll} & \text{size } t \text{ volume sampling from } \mathbf{X} & T \overset{t}{\sim} \mathbf{X} \\ & \text{size } s \text{ volume sampling from } \mathbf{X}_T & S \overset{s}{\sim} \mathbf{X}_T \\ \hline = & \text{size } s \text{ volume sampling from } \mathbf{X} & = S \overset{s}{\sim} \mathbf{X} \end{array}$$

Expectation formulas for Reverse Iterative Sampling

To each subset S assign a formula $\mathbf{F}(S)$

Property:

$$\text{If} \quad \overbrace{\sum_{i \in S} P(i|S)\mathbf{F}(S_{-i})}^{\mathbb{E}_i[\mathbf{F}(S_{-i}) \mid S]} = \mathbf{F}(S) \quad \text{for all } S,$$

$$\text{then} \quad \mathbb{E}_S[\mathbf{F}(S)] = \mathbf{F}(\{1..n\})$$

Examples of expectation formulas

1. Pseudo-inverse formula for size s volume sampling:

$$\mathbb{E}[(\mathbf{I}_S\mathbf{X})^+] = \mathbf{X}^+, \qquad \text{for any } s \geq d$$

Implies: that $\mathbf{w}^*(S)$ is an unbiased estimator of $\mathbf{w}^*$

2. Square-inverse formula for size s volume sampling:

$$\mathbb{E}[\underbrace{(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}}_{(\mathbf{I}_S\mathbf{X})^+(\mathbf{I}_S\mathbf{X})^{+\top}}] = \frac{n-d+1}{s-d+1}\underbrace{(\mathbf{X}^\top\mathbf{X})^{-1}}_{\mathbf{X}^+\mathbf{X}^{+\top}}$$

Implies: a variance bound on $\mathbf{w}^*(S)$

Examples of expectation formulas

1. Pseudo-inverse formula for size s volume sampling:

$$\mathbb{E}[(\mathbf{I}_S\mathbf{X})^+] = \mathbf{X}^+, \qquad \text{for any } s \geq d$$

Implies: that $\mathbf{w}^*(S)$ is an unbiased estimator of $\mathbf{w}^*$

2. Square-inverse formula for size s volume sampling:

$$\mathbb{E}[\underbrace{(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}}_{(\mathbf{I}_S\mathbf{X})^+(\mathbf{I}_S\mathbf{X})^{+\top}}] = \frac{n-d+1}{s-d+1}\underbrace{(\mathbf{X}^\top\mathbf{X})^{-1}}_{\mathbf{X}^+\mathbf{X}^{+\top}}$$

Implies: a variance bound on $\mathbf{w}^*(S)$

Example 1: Proof of $\mathbb{E}[(\mathbf{I}_S\mathbf{X})^+] = \mathbf{X}^+$

Suffices to show

$$\mathbf{F}(S) = \overbrace{\sum_{i\in S} P(i|S)\mathbf{F}(S_{-i})}^{\mathbb{E}[\mathbf{F}(S_{-i})\,|\,S]}$$

$$\text{for} \quad P(i|S) = \frac{1 - \mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i}{s-d} \quad \text{and} \quad \mathbf{F}(S) = (\mathbf{I}_S\mathbf{X})^+$$

This becomes:

$$(\mathbf{I}_S\mathbf{X})^+ = \sum_{i\in S} \frac{1 - \mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i}{s-d} \underbrace{(\mathbf{I}_{S_{-i}}\mathbf{X})^+}_{(\mathbf{X}_{S_{-i}}^\top\mathbf{X}_{S_{-i}})^{-1}(\mathbf{I}_{S_{-i}}\mathbf{X})^\top}$$

Proven by applying Sherman Morrison formula

$$\overbrace{(\mathbf{X}_S^\top\mathbf{X}_S - \mathbf{x}_i\mathbf{x}_i^\top)^{-1}}^{(\mathbf{X}_{S_{-i}}^\top\mathbf{X}_{S_{-i}})^{-1}} = (\mathbf{X}_S^\top\mathbf{X}_S)^{-1} + \frac{(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}}{1-\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i}$$

Example 1: Proof of $\mathbb{E}[(\mathbf{I}_S\mathbf{X})^+] = \mathbf{X}^+$

Suffices to show

$$\mathbf{F}(S) = \overbrace{\sum_{i\in S} P(i|S)\mathbf{F}(S_{-i})}^{\mathbb{E}[\mathbf{F}(S_{-i})\,|\,S]}$$

$$\text{for} \quad P(i|S) = \frac{1 - \mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i}{s-d} \quad \text{and} \quad \mathbf{F}(S) = (\mathbf{I}_S\mathbf{X})^+$$

This becomes:

$$(\mathbf{I}_S\mathbf{X})^+ = \sum_{i\in S} \frac{1 - \mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i}{s-d} \underbrace{(\mathbf{I}_{S_{-i}}\mathbf{X})^+}_{(\mathbf{X}_{S_{-i}}^\top\mathbf{X}_{S_{-i}})^{-1}(\mathbf{I}_{S_{-i}}\mathbf{X})^\top}$$

Proven by applying Sherman Morrison formula

$$\overbrace{(\mathbf{X}_S^\top\mathbf{X}_S - \mathbf{x}_i\mathbf{x}_i^\top)^{-1}}^{(\mathbf{X}_{S_{-i}}^\top\mathbf{X}_{S_{-i}})^{-1}} = (\mathbf{X}_S^\top\mathbf{X}_S)^{-1} + \frac{(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}}{1-\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i}$$

Example 1: Proof of $\mathbb{E}[(\mathbf{I}_S\mathbf{X})^+] = \mathbf{X}^+$

Suffices to show

$$\mathbf{F}(S) = \overbrace{\sum_{i\in S} P(i|S)\mathbf{F}(S_{-i})}^{\mathbb{E}[\mathbf{F}(S_{-i})\,|\,S]}$$

$$\text{for} \quad P(i|S) = \frac{1 - \mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i}{s-d} \quad \text{and} \quad \mathbf{F}(S) = (\mathbf{I}_S\mathbf{X})^+$$

This becomes:

$$(\mathbf{I}_S\mathbf{X})^+ = \sum_{i\in S} \frac{1 - \mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i}{s-d} \underbrace{(\mathbf{I}_{S_{-i}}\mathbf{X})^+}_{(\mathbf{X}_{S_{-i}}^\top\mathbf{X}_{S_{-i}})^{-1}(\mathbf{I}_{S_{-i}}\mathbf{X})^\top}$$

Proven by applying Sherman Morrison formula

$$\overbrace{(\mathbf{X}_S^\top\mathbf{X}_S - \mathbf{x}_i\mathbf{x}_i^\top)^{-1}}^{(\mathbf{X}_{S_{-i}}^\top\mathbf{X}_{S_{-i}})^{-1}} = (\mathbf{X}_S^\top\mathbf{X}_S)^{-1} + \frac{(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}}{1-\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i}$$

Example 2: Proof of $\mathbb{E}[(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}] = \frac{n-d+1}{s-d+1}(\mathbf{X}^\top \mathbf{X})^{-1}$

Use $\mathbf{F}(S) = \frac{s-d+1}{n-d+1}(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}$.

Show $\mathbf{F}(S) = \sum_{i \in S} P(i|S)\mathbf{F}(S_{-i})$ for volume sampling, i.e.

$$\cancel{\frac{s-d+1}{n-d+1}}(\mathbf{X}_S^\top \mathbf{X}_S)^{-1} = \sum_{i \in S} \frac{1 - \mathbf{x}_i^\top (\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{x}_i}{\cancel{s-d}} \frac{\cancel{s-d}}{\cancel{n-d+1}}(\mathbf{X}_{S_{-i}}^\top \mathbf{X}_{S_{-i}})^{-1}$$

Apply Sherman Morrison to $(\mathbf{X}_{S_{-i}}^\top \mathbf{X}_{S_{-i}})^{-1}$ and simplify

Outline

Unbiased estimators are easy to combine

Simple Strategy:

1. Compute independent estimators $\mathbf{w}(S_j)$ for $j = 1, .., k$,
2. Predict with the average estimator $\frac{1}{k}\sum_{j=1}^{k}\mathbf{w}(S_j)$

If we have

$$\mathbb{E}[L(\mathbf{w}(S))] \leq (1+c)L(\mathbf{w}^*) \quad \text{and} \quad \mathbb{E}[\mathbf{w}(S)] = \mathbf{w}^*,$$

then for k independent samples $S_1, \ldots, S_k$,

$$\mathbb{E}\left[L\Big(\frac{1}{k}\sum_{j=1}^{k}\mathbf{w}(S_j)\Big)\right] \leq \Big(1+\frac{c}{k}\Big)\,L(\mathbf{w}^*)$$

Motivation:

- Ensemble methods
- Distributed optimization
- Privacy

Unbiased estimators are easy to combine

Simple Strategy:

1. Compute independent estimators $\mathbf{w}(S_j)$ for $j = 1, .., k$,
2. Predict with the average estimator $\frac{1}{k}\sum_{j=1}^{k}\mathbf{w}(S_j)$

If we have

$$\mathbb{E}[L(\mathbf{w}(S))] \leq (1+c)L(\mathbf{w}^*) \quad \text{and} \quad \mathbb{E}[\mathbf{w}(S)] = \mathbf{w}^*,$$

then for k independent samples $S_1, \ldots, S_k$,

$$\mathbb{E}\left[L\left(\frac{1}{k}\sum_{j=1}^{k}\mathbf{w}(S_j)\right)\right] \leq \left(1+\frac{c}{k}\right)L(\mathbf{w}^*)$$

Motivation:

- Ensemble methods
- Distributed optimization
- Privacy

Unbiased estimators are easy to combine

Simple Strategy:

1. Compute independent estimators $\mathbf{w}(S_j)$ for $j = 1, .., k$,
2. Predict with the average estimator $\frac{1}{k}\sum_{j=1}^{k}\mathbf{w}(S_j)$

If we have

$$\mathbb{E}[L(\mathbf{w}(S))] \leq (1+c)L(\mathbf{w}^*) \quad \text{and} \quad \mathbb{E}[\mathbf{w}(S)] = \mathbf{w}^*,$$

then for k independent samples $S_1, \ldots, S_k$,

$$\mathbb{E}\left[L\left(\frac{1}{k}\sum_{j=1}^{k}\mathbf{w}(S_j)\right)\right] \leq \left(1+\frac{c}{k}\right)L(\mathbf{w}^*)$$

Motivation:

- Ensemble methods
- Distributed optimization
- Privacy

Open problems

Open: Is there a size $O(d/\epsilon)$ unbiased estimator that achieves

$$\mathbb{E}\big[L(\mathbf{w}(S))\big] \leq (1+\epsilon)L(\mathbf{w}^*) \ ?$$

Equivalent: Is there a size $O(d)$ unbiased estimator that achieves a constant factor approximation?

By averaging $1/\epsilon$ copies, we would get the $1 + O(\epsilon)$ approximation

Our progress so far:

1. size $O(d^2/\epsilon)$ (averaging size d volume sampling) [DW17]
2. size $O(d/\epsilon)$ (only if $\mathbf{y}$ is linear plus white noise) [DW18a]
3. size $O(d \log d + d/\epsilon)$ (leveraged volume sampling) [DWH18]

Open problems

Open: Is there a size $O(d/\epsilon)$ unbiased estimator that achieves

$$\mathbb{E}\big[L(\mathbf{w}(S))\big] \leq (1+\epsilon)L(\mathbf{w}^*) \text{ ?}$$

Equivalent: Is there a size $O(d)$ unbiased estimator that achieves a constant factor approximation?

By averaging $1/\epsilon$ copies, we would get the $1 + O(\epsilon)$ approximation

Our progress so far:

1. size $O(d^2/\epsilon)$ (averaging size d volume sampling) [DW17]
2. size $O(d/\epsilon)$ (only if $\mathbf{y}$ is linear plus white noise) [DW18a]
3. size $O(d \log d + d/\epsilon)$ (leveraged volume sampling) [DWH18]

Open problems

Open: Is there a size $O(d/\epsilon)$ unbiased estimator that achieves

$$\mathbb{E}\big[L(\mathbf{w}(S))\big] \leq (1+\epsilon)L(\mathbf{w}^*) \ ?$$

Equivalent: Is there a size $O(d)$ unbiased estimator that achieves a constant factor approximation?

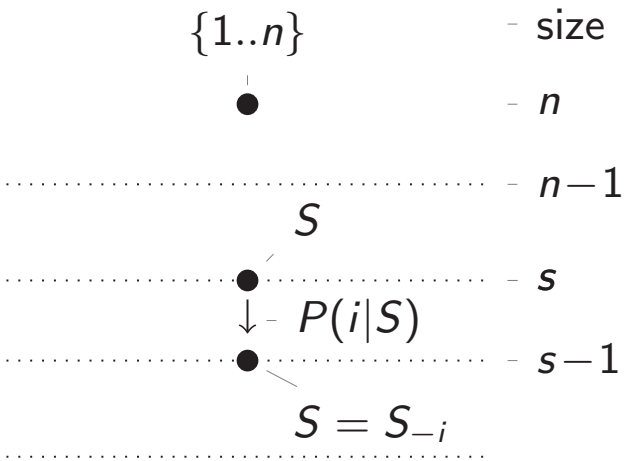
By averaging $1/\epsilon$ copies, we would get the $1 + O(\epsilon)$ approximation

Our progress so far:

1. size $O(d^2/\epsilon)$ (averaging size d volume sampling) [DW17]
2. size $O(d/\epsilon)$ (only if $\mathbf{y}$ is linear plus white noise) [DW18a]
3. size $O(d \log d + d/\epsilon)$ (leveraged volume sampling) [DWH18]

Outline

Simple algorithm for volume sampling



Start with $S = \{1..n\}$

Sample index $i \in S$

Go to set $S_{-i} = S - \{i\}$

Repeat until desired size

Simple algorithm: Update distribution $P(i|S)$ at every step

Runtime: $\overbrace{\text{steps}}^{n-s} \times \overbrace{\text{update}}^{O(n\,d)} = O(n^2 d)$

Problem: Quadratic dependence on n

Faster algorithm via rejection sampling

Recall: $P(i|S) \sim 1 - \mathbf{x}_i^\top (\mathbf{X}_S^\top \mathbf{X}_S)^{-1} \mathbf{x}_i$

Idea: Rejection sampling from distribution $P(i|S)$

1. Sample i uniformly from set S,
2. Compute $h_i = 1 - \mathbf{x}_i^\top (\mathbf{X}_S^\top \mathbf{X}_S)^{-1} \mathbf{x}_i$,
3. With probability $1 - h_i$ reject and go back to 1.

one trial

We show: Number of trials per step is constant w.h.p.

Runtime: $\overbrace{\text{steps}}^{n-s} \times \overbrace{\text{trials per step}}^{O(1)} \times \overbrace{\text{compute } h_i}^{O(d^2)} = O(nd^2)$

Result: Linear dependence on n

Faster algorithm via rejection sampling

Recall: $P(i|S) \sim 1 - \mathbf{x}_i^\top (\mathbf{X}_S^\top \mathbf{X}_S)^{-1} \mathbf{x}_i$

Idea: Rejection sampling from distribution $P(i|S)$

1. Sample i uniformly from set S,
2. Compute $h_i = 1 - \mathbf{x}_i^\top (\mathbf{X}_S^\top \mathbf{X}_S)^{-1} \mathbf{x}_i$,
3. With probability $1 - h_i$ reject and go back to 1.

} one trial

We show: Number of trials per step is constant w.h.p.

Runtime: $\overbrace{\text{steps}}^{n-s} \times \overbrace{\text{trials per step}}^{O(1)} \times \overbrace{\text{compute } h_i}^{O(d^2)} = O(nd^2)$

Result: Linear dependence on n

Extension: regularized volume sampling [DW18a]

Goal: Error bounds for sets of size $\ll d$

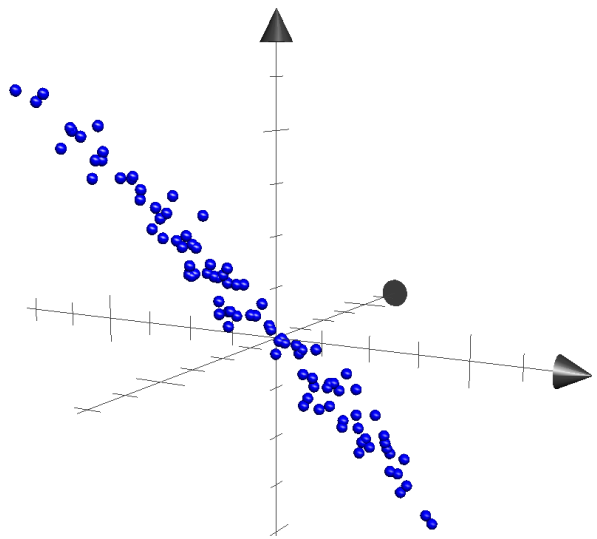
Instead of $\det(\mathbf{X}_S^\top \mathbf{X}_S)$

use $\det(\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})$

λ-statistical dimension

$\lambda_1, \ldots, \lambda_d$ - eigenvalues of $\mathbf{X}^\top \mathbf{X}$

$$d_\lambda = \sum_{i=1}^{d} \frac{\lambda_i}{\lambda_i + \lambda} \; < \; d$$



Result: With properly tuned λ, it suffices to sample d_λ labels

Extension: regularized volume sampling [DW18a]

Goal: Error bounds for sets of size $\ll d$

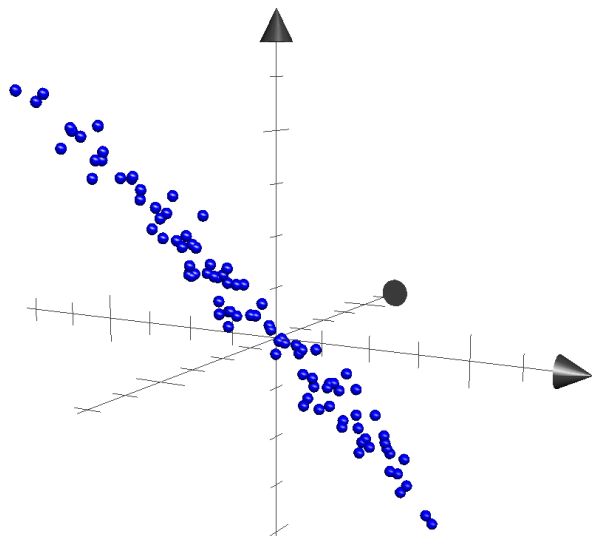
Instead of $\det(\mathbf{X}_S^\top \mathbf{X}_S)$

use $\det(\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})$

λ-statistical dimension

$\lambda_1, \ldots, \lambda_d$ - eigenvalues of $\mathbf{X}^\top \mathbf{X}$

$$d_\lambda = \sum_{i=1}^{d} \frac{\lambda_i}{\lambda_i + \lambda} \; < \; d$$



Result: With properly tuned λ, it suffices to sample d_λ labels

Extension: regularized volume sampling [DW18a]

Goal: Error bounds for sets of size $\ll d$

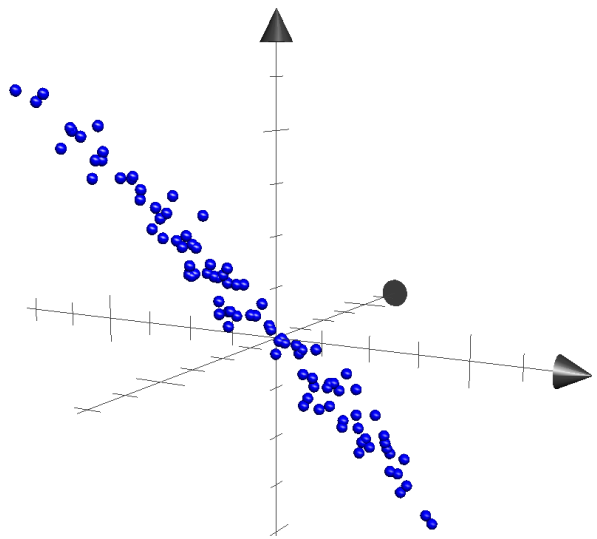
$$\text{Instead of} \qquad \det(\mathbf{X}_S^\top \mathbf{X}_S)$$

$$\text{use} \qquad \det(\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})$$

λ-statistical dimension

$\lambda_1, \ldots, \lambda_d$ - eigenvalues of $\mathbf{X}^\top \mathbf{X}$

$$d_\lambda = \sum_{i=1}^{d} \frac{\lambda_i}{\lambda_i + \lambda} \; < \; d$$



Result: With properly tuned λ, it suffices to sample d_λ labels

Extension: regularized volume sampling [DW18a]

Goal: Error bounds for sets of size $\ll d$

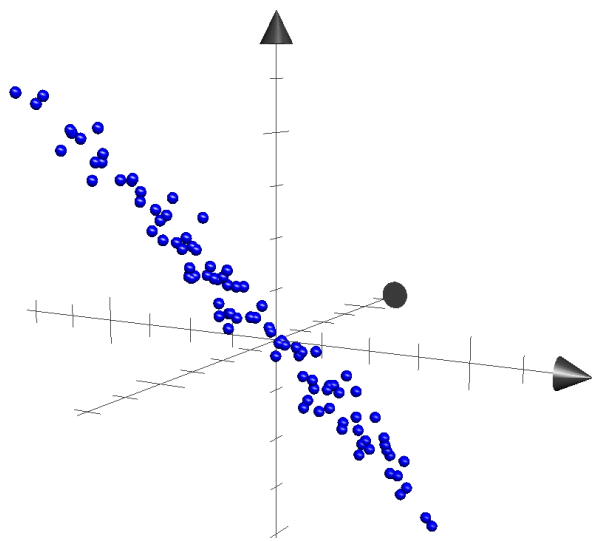
$$\text{Instead of} \qquad \det(\mathbf{X}_S^\top \mathbf{X}_S)$$

$$\text{use} \qquad \det(\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})$$

λ-statistical dimension

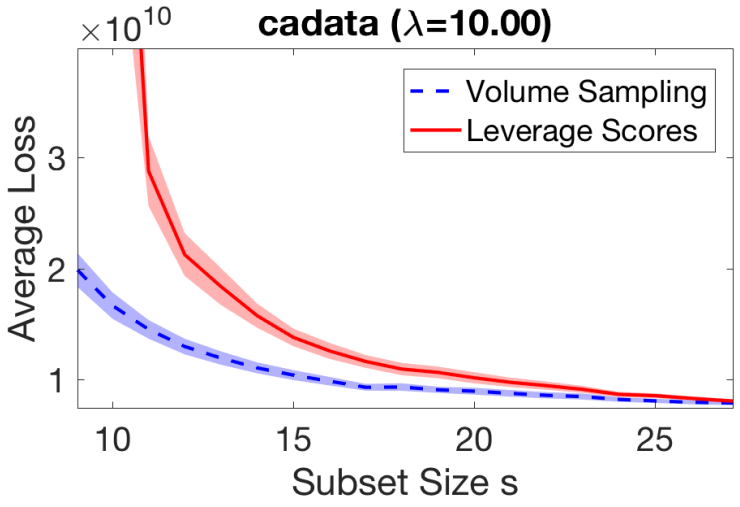
$\lambda_1, \ldots, \lambda_d$ - eigenvalues of $\mathbf{X}^\top \mathbf{X}$

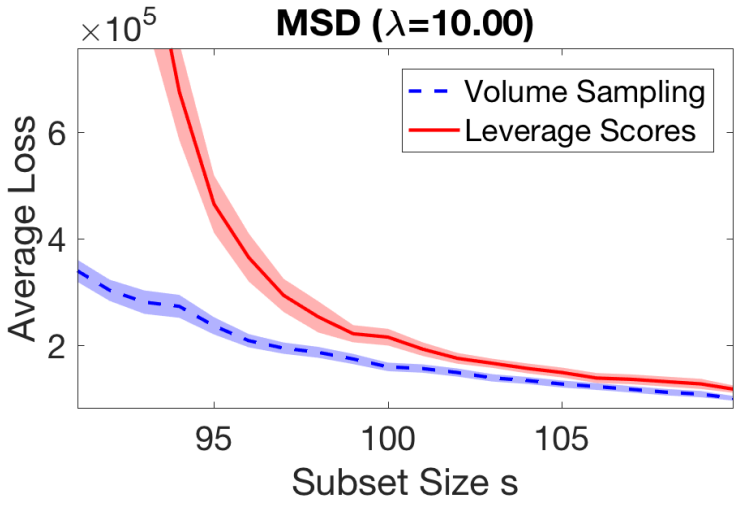
$$d_\lambda = \sum_{i=1}^{d} \frac{\lambda_i}{\lambda_i + \lambda} \; < \; d$$

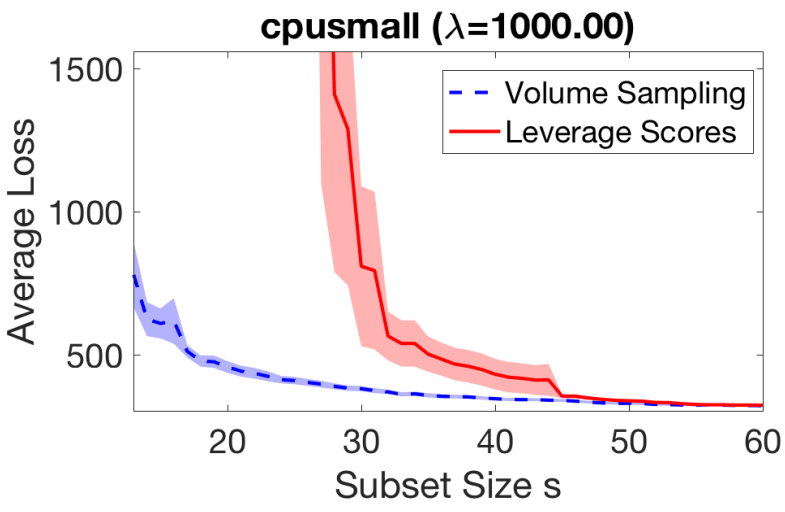


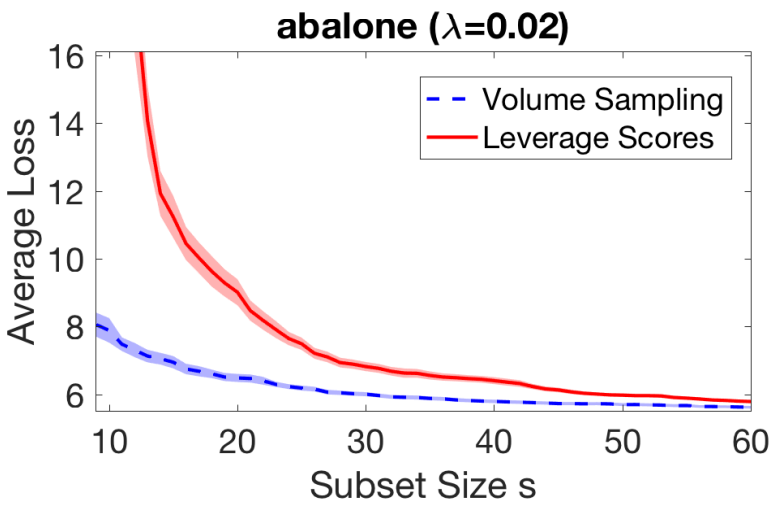
Result: With properly tuned λ, it suffices to sample d_λ labels

Regularized volume sampling vs iid leverage scores









λ indicates the amount of ℓ_2-regularization used for sampling and prediction

Extension: rescaled volume sampling [DWH18]

Instead of $\det\Big(\overbrace{\sum_{i\in S}\mathbf{x}_i\mathbf{x}_i^\top}^{\mathbf{x}_S^\top\mathbf{x}_S}\Big)$

use $\det\Big(\sum_{t=1}^{s}\frac{1}{q_{i_t}}\mathbf{x}_{i_t}\mathbf{x}_{i_t}^\top\Big)\prod_{t=1}^{s}q_{i_t}$

New family of unbiased least squares estimators:

Leveraged volume sampling: better bounds for unbiased estimator

Determinantal rejection sampling: faster algorithm

Extension: rescaled volume sampling [DWH18]

$$\text{Instead of} \qquad \det\Big(\overbrace{\sum_{i\in S}\mathbf{x}_i\mathbf{x}_i^\top}^{\mathbf{x}_S^\top\mathbf{x}_S}\Big)$$

$$\text{use} \qquad \det\Big(\sum_{t=1}^{s}\frac{1}{q_{i_t}}\mathbf{x}_{i_t}\mathbf{x}_{i_t}^\top\Big)\prod_{t=1}^{s}q_{i_t}$$

New family of unbiased least squares estimators:

Leveraged volume sampling: better bounds for unbiased estimator

Determinantal rejection sampling: faster algorithm

Extension: rescaled volume sampling [DWH18]

$$\text{Instead of} \qquad \det\Big(\overbrace{\sum_{i\in S}\mathbf{x}_i\mathbf{x}_i^\top}^{\mathbf{x}_S^\top\mathbf{x}_S}\Big)$$

$$\text{use} \qquad \det\Big(\sum_{t=1}^{s}\frac{1}{q_{i_t}}\mathbf{x}_{i_t}\mathbf{x}_{i_t}^\top\Big)\prod_{t=1}^{s}q_{i_t}$$

New family of unbiased least squares estimators:

Leveraged volume sampling: better bounds for unbiased estimator

Determinantal rejection sampling: faster algorithm

Extension: rescaled volume sampling [DWH18]

$$\text{Instead of} \qquad \det\Big(\overbrace{\sum_{i\in S}\mathbf{x}_i\mathbf{x}_i^\top}^{\mathbf{x}_S^\top\mathbf{x}_S}\Big)$$

$$\text{use} \qquad \det\Big(\sum_{t=1}^{s}\frac{1}{q_{i_t}}\mathbf{x}_{i_t}\mathbf{x}_{i_t}^\top\Big)\prod_{t=1}^{s}q_{i_t}$$

New family of unbiased least squares estimators:

Leveraged volume sampling: better bounds for unbiased estimator

Determinantal rejection sampling: faster algorithm

Leveraged volume sampling

- rescaled volume sampling
- with iid leverage scores:

$$q_i \sim \mathbf{x}_i^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_i$$

Determinantal rejection sampling trick

repeat

Sample $i_1, \ldots, i_s$ i.i.d. $\sim (q_1, \ldots, q_n)$

Sample *Accept* $\sim$ Bernoulli $\left[\frac{\det\left(\sum_{t=1}^{s} \frac{1}{q_{i_t}} \mathbf{x}_{i_t} \mathbf{x}_{i_t}^\top \right)}{\det(\mathbf{X}^\top \mathbf{X})} \right]$

until *Accept* $=$ true

$$\underbrace{\text{preprocessing } O(nd^2)}_{\text{improvable to } \widetilde{O}(nd+d^3)} + \underbrace{\text{sampling } O(d^4)}_{\text{no dependence on } n}$$

Removes the bias from iid leverage score sampling!

Approximate leverage scores: $\widetilde{O}(nd + d^3)$

First polynomial alg. for volume sample: $O(n^4 s)$ [LJS17]

Leveraged volume sampling

- rescaled volume sampling
- with iid leverage scores:

$$q_i \sim \mathbf{x}_i^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_i$$

Determinantal rejection sampling trick

repeat

Sample $i_1, \ldots, i_s$ i.i.d. $\sim (q_1, \ldots, q_n)$

Sample *Accept* $\sim$ Bernoulli $\left[\frac{\det\left(\sum_{t=1}^{s} \frac{1}{q_{i_t}} \mathbf{x}_{i_t} \mathbf{x}_{i_t}^\top \right)}{\det(\mathbf{X}^\top \mathbf{X})} \right]$

until *Accept* $=$ true

$$\underbrace{\text{preprocessing } O(nd^2)}_{\text{improvable to } \widetilde{O}(nd+d^3)} + \underbrace{\text{sampling } O(d^4)}_{\text{no dependence on } n}$$

Removes the bias from iid leverage score sampling!

Approximate leverage scores: $\widetilde{O}(nd + d^3)$

First polynomial alg. for volume sample: $O(n^4 s)$ [LJS17]

Leveraged volume sampling

- rescaled volume sampling
- with iid leverage scores:

$$q_i \sim \mathbf{x}_i^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_i$$

Determinantal rejection sampling trick

repeat

Sample $i_1, \ldots, i_s$ i.i.d. $\sim (q_1, \ldots, q_n)$

Sample *Accept* $\sim$ Bernoulli $\left[\frac{\det\left(\sum_{t=1}^{s} \frac{1}{q_{i_t}} \mathbf{x}_{i_t} \mathbf{x}_{i_t}^\top \right)}{\det(\mathbf{X}^\top \mathbf{X})} \right]$

until *Accept* $=$ true

$$\underbrace{\text{preprocessing } O(nd^2)}_{\text{improvable to } \widetilde{O}(nd+d^3)} + \underbrace{\text{sampling } O(d^4)}_{\text{no dependence on } n}$$

Removes the bias from iid leverage score sampling!

Approximate leverage scores: $\widetilde{O}(nd + d^3)$

First polynomial alg. for volume sample: $O(n^4 s)$ [LJS17]

Leveraged volume sampling

- rescaled volume sampling
- with iid leverage scores:

$$q_i \sim \mathbf{x}_i^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_i$$

Determinantal rejection sampling trick

repeat

Sample $i_1, \ldots, i_s$ i.i.d. $\sim (q_1, \ldots, q_n)$

Sample *Accept* $\sim$ Bernoulli $\left[\frac{\det\left(\sum_{t=1}^{s} \frac{1}{q_{i_t}} \mathbf{x}_{i_t} \mathbf{x}_{i_t}^\top \right)}{\det(\mathbf{X}^\top \mathbf{X})} \right]$

until *Accept* $=$ true

$$\underbrace{\text{preprocessing } O(nd^2)}_{\text{improvable to } \widetilde{O}(nd+d^3)} + \underbrace{\text{sampling } O(d^4)}_{\text{no dependence on } n}$$

Removes the bias from iid leverage score sampling!

Approximate leverage scores: $\widetilde{O}(nd + d^3)$

First polynomial alg. for volume sample: $O(n^4 s)$ [LJS17]

Efficiency of volume sampling

Libsvm dataset: YearPredictionMSD ($n = 463715$, $d = 90$)

computing leverage scores: (preprocessing step)

volume sampling	**runtime**	
previous state-of-the-art:		[LJS17] Mar. 2017
our volume sampling algorithms		
reverse iterative sampling:		[DW17] May 2017
fast reverse iterative sampling:		[DW18a] Oct. 2017
leveraged volume sampling:		[DWH18] May 2018

Efficiency of volume sampling

Libsvm dataset: YearPredictionMSD ($n = 463715$, $d = 90$)

computing leverage scores: 10 seconds (preprocessing step)

volume sampling **runtime**

previous state-of-the-art: [LJS17] Mar. 2017

our volume sampling algorithms

reverse iterative sampling: [DW17] May 2017

fast reverse iterative sampling: [DW18a] Oct. 2017

leveraged volume sampling: [DWH18] May 2018

Efficiency of volume sampling

Libsvm dataset: YearPredictionMSD ($n = 463715$, $d = 90$)

computing leverage scores: 10 seconds (preprocessing step)

volume sampling	**runtime**	
previous state-of-the-art:	age of the universe	[LJS17] Mar. 2017
our volume sampling algorithms		
reverse iterative sampling:		[DW17] May 2017
fast reverse iterative sampling:		[DW18a] Oct. 2017
leveraged volume sampling:		[DWH18] May 2018

Efficiency of volume sampling

Libsvm dataset: YearPredictionMSD ($n = 463715$, $d = 90$)

computing leverage scores: 10 seconds (preprocessing step)

volume sampling	**runtime**	
previous state-of-the-art:	age of the universe	[LJS17] Mar. 2017
our volume sampling algorithms		
reverse iterative sampling:	24 hours	[DW17] May 2017
fast reverse iterative sampling:		[DW18a] Oct. 2017
leveraged volume sampling:		[DWH18] May 2018

Efficiency of volume sampling

Libsvm dataset: YearPredictionMSD ($n = 463715$, $d = 90$)

computing leverage scores: 10 seconds (preprocessing step)

volume sampling	**runtime**	
previous state-of-the-art:	age of the universe	[LJS17] Mar. 2017
our volume sampling algorithms		
reverse iterative sampling:	24 hours	[DW17] May 2017
fast reverse iterative sampling:	30 seconds	[DW18a] Oct. 2017
leveraged volume sampling:		[DWH18] May 2018

Efficiency of volume sampling

Libsvm dataset: YearPredictionMSD ($n = 463715$, $d = 90$)

computing leverage scores: 10 seconds (preprocessing step)

volume sampling	**runtime**	
previous state-of-the-art:	age of the universe	[LJS17] Mar. 2017
our volume sampling algorithms		
reverse iterative sampling:	24 hours	[DW17] May 2017
fast reverse iterative sampling:	30 seconds	[DW18a] Oct. 2017
leveraged volume sampling:	3 seconds	[DWH18] May 2018

Conclusion

We showed

- Volume sampling is fundamental, elegant, quite fast
- Leads to unbiased estimators

And what is next?

- Subsampled Newton's method
- Applications in distributed computing
- What part of this generalizes to tensors?

Conclusion

We showed

- Volume sampling is fundamental, elegant, quite fast
- Leads to unbiased estimators

And what is next?

- Subsampled Newton's method
- Applications in distributed computing
- What part of this generalizes to tensors?

References

[DW17] Dereziński, Warmuth. *Unbiased estimates for linear regression via volume sampling*. NIPS 2017.

[DW18a] Dereziński, Warmuth. *Subsampling for ridge regression via regularized volume sampling*. AISTATS 2018.

[DW18b] Dereziński, Warmuth. *Reverse iterative volume sampling for linear regression*. JMLR (to appear), 2018.

[DWH18] Dereziński, Warmuth, Hsu. *Leveraged volume sampling for linear regression*. 2018.
`https://arxiv.org/abs/1802.06749`