Unbiased estimates for linear regression via volume sampling

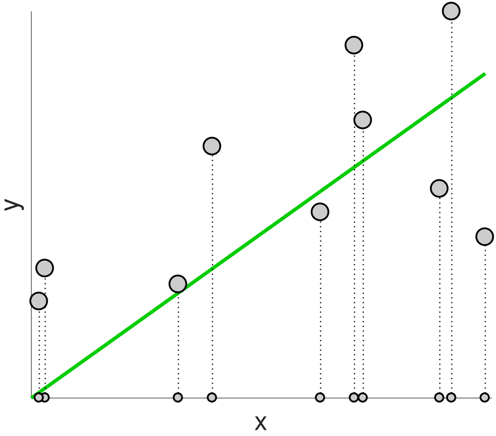
Michał Dereziński
UC Berkeley

Joint work with Manfred Warmuth, Daniel Hsu

Simons Institute, September 26, 2018

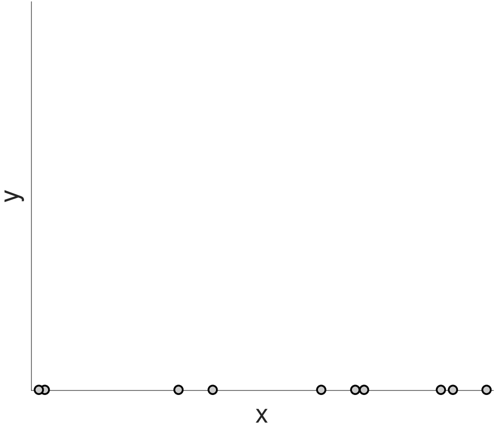
Outline

Least squares regression



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

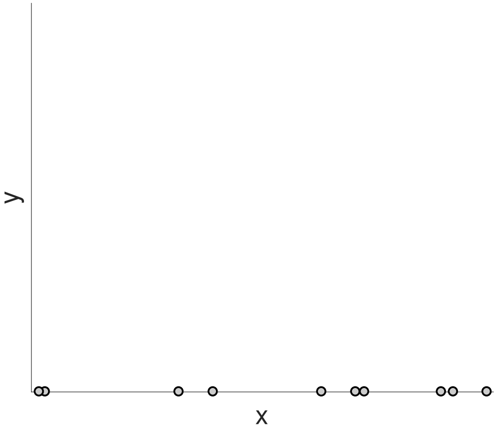
How many labels needed to get close to optimum?



- All x_i given
- But labels y_i unknown

Guess how many needed?

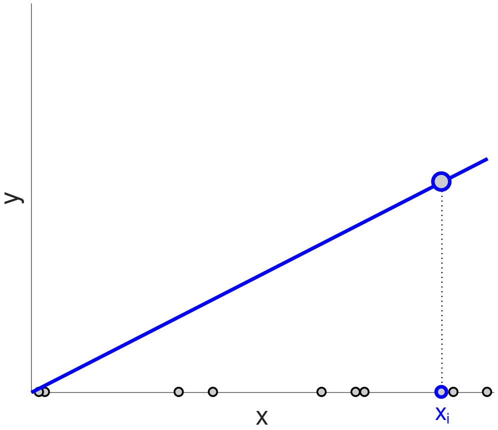
How many labels needed to get close to optimum?



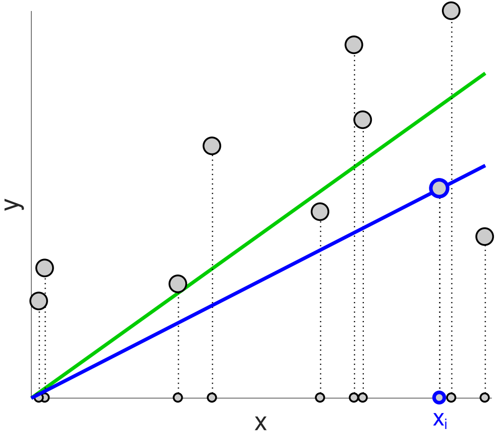
- All x_i given
- But labels y_i unknown

Guess how many needed?

Answer: one label

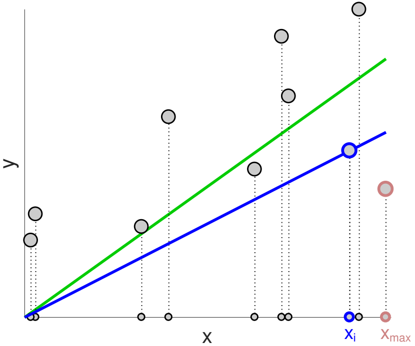


How good is one label?



$\color{blue}{\text{Loss of estimate}} = \color{red}{2} \times \color{green}{\text{Loss of optimum}}$

Which one?



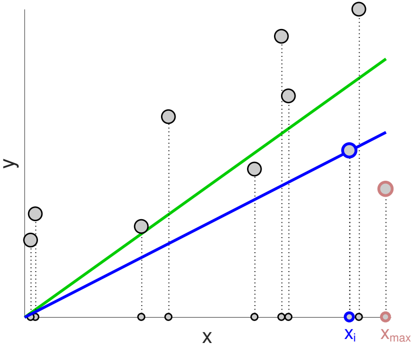
- $x_{\max}$ (furthest from 0) is bad
- any deterministic choice is bad

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?



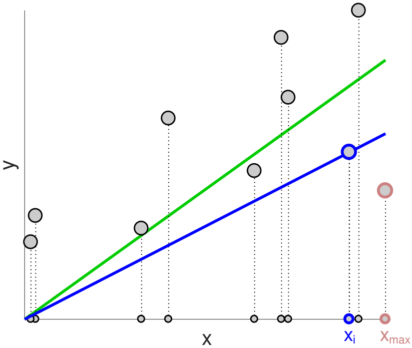
- x_{max} (furthest from 0) is bad
- any deterministic choice is bad

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?



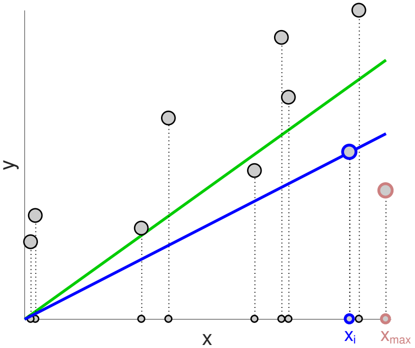
- x_{max} (furthest from 0) is bad
- any deterministic choice is bad

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

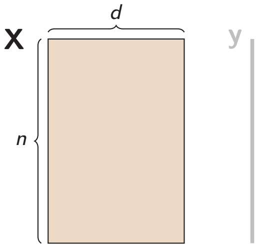
Good: 1 label y_i drawn $\sim x_i^2$

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

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

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:

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

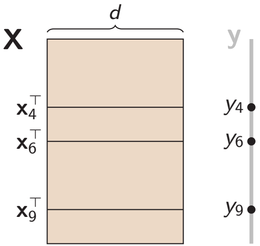
$(\mathbf{X}_S)^+ = (\mathbf{X}_S^\top \mathbf{X}_S)^{-1} \mathbf{X}_S^\top$ - pseudo-inverse of $\mathbf{X}_S$

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:

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

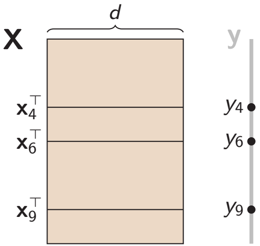
$(\mathbf{X}_S)^+ = (\mathbf{X}_S^\top \mathbf{X}_S)^{-1} \mathbf{X}_S^\top$ - pseudo-inverse of $\mathbf{X}_S$

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:

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

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

Volume Sampling [DRVW06, AB13]

For $d = 1$*:* pick set $S = \{i\}$ w.p. $P(S) \propto x_i^2$

For any d*:* pick d-element S w.p. $P(S) \propto \det(\mathbf{X}_S)^2$

Distribution over all s-element subsets S (for fixed $s \geq d$):

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

Normalization factor Z is derived using Cauchy-Binet formula:

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

Volume Sampling [DRVW06, AB13]

For $d = 1$*:* pick set $S = \{i\}$ w.p. $P(S) \propto x_i^2$

For any d*:* pick d-element S w.p. $P(S) \propto \det(\mathbf{X}_S)^2$

Distribution over all s-element subsets S (for fixed $s \geq d$):

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

Normalization factor Z is derived using Cauchy-Binet formula:

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

Volume Sampling [DRVW06, AB13]

For $d=1$*:* pick set $S=\{i\}$ w.p. $P(S) \propto x_i^2$

For any d*:* pick d-element S w.p. $P(S) \propto \det(\mathbf{X}_S)^2$

Distribution over all s-element subsets S (for fixed $s \geq d$):

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

Normalization factor Z is derived using Cauchy-Binet formula:

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

Volume Sampling [DRVW06, AB13]

For $d = 1$*:* pick set $S = \{i\}$ w.p. $P(S) \propto x_i^2$

For any d*:* pick d-element S w.p. $P(S) \propto \det(\mathbf{X}_S)^2$

Distribution over all s-element subsets S (for fixed $s \geq d$):

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

Normalization factor Z is derived using Cauchy-Binet formula:

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

Outline

Linear regression with dimension many labels

Theorem ([DW17])

For a volume-sampled d-element set S,

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

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

- Sampling distribution does not depend on the labels
- No range restrictions

What about iid leverage score sampling?

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 (coupon collector problem)
2. produces biased estimators

Volume sampling

Requires only d labels and produces unbiased estimators

What about iid leverage score sampling?

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 $\underline{d \log d}$ labels (coupon collector problem)
2. produces <u>biased</u> estimators

Volume sampling

Requires only d labels and produces <u>unbiased</u> estimators

What about iid leverage score sampling?

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 $\underline{d \log d}$ labels (coupon collector problem)
2. produces <u>biased</u> estimators

Volume sampling

Requires only d labels and produces <u>unbiased</u> estimators

What about iid leverage score sampling?

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

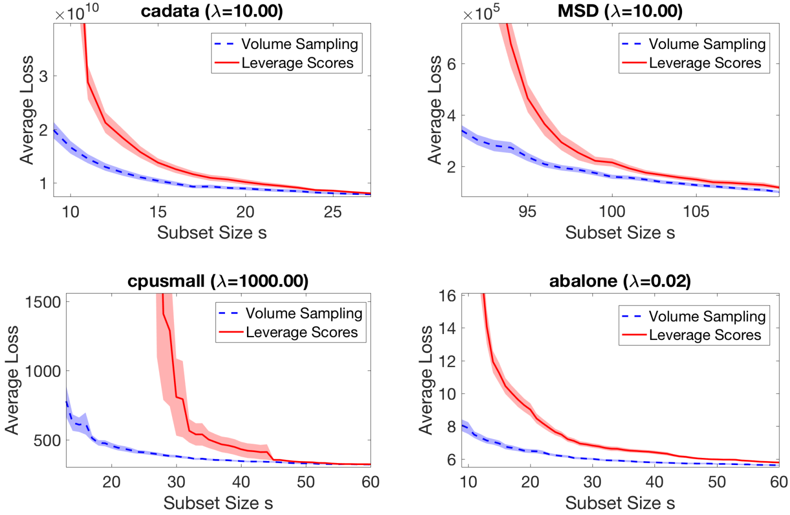
Problems with iid sampling:

1. requires at least $\underline{d \log d}$ labels (coupon collector problem)
2. produces <u>biased</u> estimators

Volume sampling

Requires only d labels and produces <u>unbiased</u> estimators

Volume sampling vs iid leverage scores



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

Why volume sampling?

- New loss bounds
  which avoid coupon collector problem
- New expectation formulas
  can be extended to matrix identities
- Unbiased estimators
  easy to combine via averaging
- Surprising closure properties
  volume sampling is closed under:
  1. subsampling
  2. adding uniform/i.i.d. samples

Why volume sampling?

- New loss bounds
  which avoid coupon collector problem
- New expectation formulas
  can be extended to matrix identities
- Unbiased estimators
  easy to combine via averaging
- Surprising closure properties

  volume sampling is closed under:
  1. subsampling
  2. adding uniform/i.i.d. samples

Why volume sampling?

- New loss bounds
  which avoid coupon collector problem
- New expectation formulas
  can be extended to matrix identities
- Unbiased estimators
  easy to combine via averaging
- Surprising closure properties
  volume sampling is closed under:
  1. subsampling
  2. adding uniform/i.i.d. samples

Why volume sampling?

- New loss bounds
  which avoid coupon collector problem

- New expectation formulas
  can be extended to matrix identities

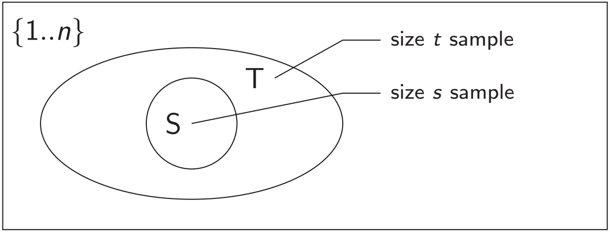
- Unbiased estimators
  easy to combine via averaging

- Surprising closure properties

  volume sampling is closed under:

  1. subsampling
  2. adding uniform/i.i.d. samples

Volume sampling is closed under subsampling

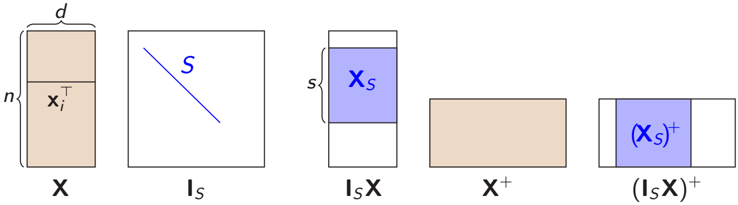


Hierarchical sampling ($t \geq s$):

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

Outline

Expectation formulas for the pseudoinverse



Expected pseudoinverse

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

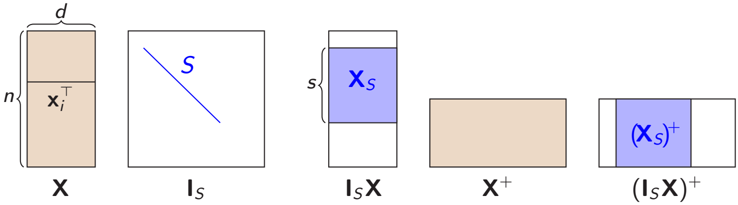
Variance of pseudoinverse estimator:

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

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

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

Expectation formulas for the pseudoinverse



Expected pseudoinverse

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

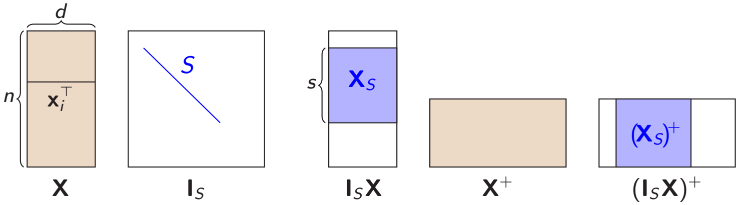
Variance of pseudoinverse estimator:

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

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

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

Expectation formulas for the pseudoinverse



Expected pseudoinverse

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

Variance of pseudoinverse estimator:

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

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

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

Expectation formulas: key technique

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

Goal: Show that $\mathbb{E}_S[\mathbf{F}(S)] = \mathbf{F}(\{1..n\})$ for size s volume sampling

Idea: Use closure under subsampling:

For fixed S of size s, sample: $S_{-i} \overset{s-1}{\sim} \mathbf{X}_S$

Suffices to show: $\mathbb{E}_i\left[\mathbf{F}(S_{-i}) \,|\, S\right] = \mathbf{F}(S)$ for all S

Example: Use formula $\mathbf{F}(S) = (\mathbf{I}_S\mathbf{X})^+$.

Follows from the Sherman-Morrison formula

Expectation formulas: key technique

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

Goal: Show that $\mathbb{E}_S[\mathbf{F}(S)] = \mathbf{F}(\{1..n\})$ for size s volume sampling

Idea: Use closure under subsampling:

For fixed S of size s, sample: $S_{-i} \overset{s-1}{\sim} \mathbf{X}_S$

Suffices to show: $\mathbb{E}_i\left[\mathbf{F}(S_{-i}) \,|\, S\right] = \mathbf{F}(S)$ for all S

Example: Use formula $\mathbf{F}(S) = (\mathbf{I}_S\mathbf{X})^+$.

Follows from the Sherman-Morrison formula

Expectation formulas: key technique

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

Goal: Show that $\mathbb{E}_S[\mathbf{F}(S)] = \mathbf{F}(\{1..n\})$ for size s volume sampling

Idea: Use closure under subsampling:

For fixed S of size s, sample: $S_{-i} \overset{s-1}{\sim} \mathbf{X}_S$

Suffices to show: $\mathbb{E}_i\big[\mathbf{F}(S_{-i}) \,|\, S\big] = \mathbf{F}(S) \quad \text{for all } S$

Example: Use formula $\mathbf{F}(S) = (\mathbf{I}_S\mathbf{X})^+$.

Follows from the Sherman-Morrison formula

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 $S_1, \ldots, S_k$,

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

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 $S_1, \ldots, S_k$,

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

Motivation:

- Ensemble methods
- Distributed optimization
- Privacy

Open problems for unbiased estimators

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

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

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]

Also biased estimators of size $O(d/\epsilon)$ are known [CP18]

Open problems for unbiased estimators

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

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

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]

Also biased estimators of size $O(d/\epsilon)$ are known [CP18]

Open problems for unbiased estimators

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

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

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]

Also biased estimators of size $O(d/\epsilon)$ are known [CP18]

Outline

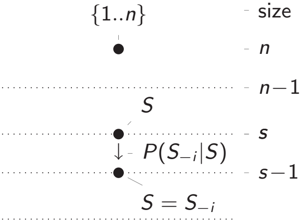
Introduction

Basic results

Unbiased estimators and matrix formulas

Algorithms and extensions

Reverse iterative 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 $P(S_{-i}|S)$ at every step

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

Problem: Quadratic dependence on n

Faster algorithm via rejection sampling

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

Idea: Rejection sampling from distribution $P(S_{-i}|S)$

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

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

Idea: Rejection sampling from distribution $P(S_{-i}|S)$

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 $\ll d$

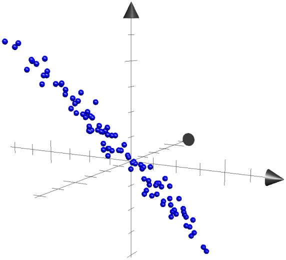
Instead of $\det(\mathbf{X}_S^\top \mathbf{X}_S)$

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 $\ll d$

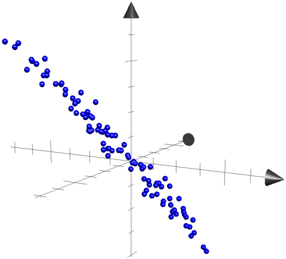
Instead of $\det(\mathbf{X}_S^\top \mathbf{X}_S)$

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: Leveraged volume sampling [DWH18]

- 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

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

Removes the bias from iid leverage score sampling!

Extension: Leveraged volume sampling [DWH18]

- 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

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

Removes the bias from iid leverage score sampling!

Extension: Leveraged volume sampling [DWH18]

- 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

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

Removes the bias from iid leverage score sampling!

Efficiency of volume sampling

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

computing leverage scores: (preprocessing step)

volume sampling **runtime**

first polynomial algorithm: [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**

first polynomial algorithm: [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**	
first polynomial algorithm:	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**	
first polynomial algorithm:	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**	
first polynomial algorithm:	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**	
first polynomial algorithm:	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?

- Introducing controlled bias into volume sampling
- Subsampled Newton's method
- Applications in distributed computing
- Connections to Determinantal Point Processes

Conclusion

We showed

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

And what is next?

- Introducing controlled bias into volume sampling
- Subsampled Newton's method
- Applications in distributed computing
- Connections to Determinantal Point Processes

References

- [DRVW06] Deshpande, Rademacher, Vempala, Wang. *Matrix approximation and projective clustering via volume sampling*. SODA 2006.
- [AB13] Avron, Boutsidis. *Faster subset selection for matrices and applications*. JMAA 2013.
- [LJS17] Li, Jegelka, Sra. *Polynomial time algorithms for dual volume sampling*. NIPS 2017.
- [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, 2018.
- [DWH18] Dereziński, Warmuth, Hsu. *Leveraged volume sampling for linear regression*. NIPS 2018 (to appear).
- [CP18] Chen, Price. *Active regression via linear-sample sparsification*. 2018.