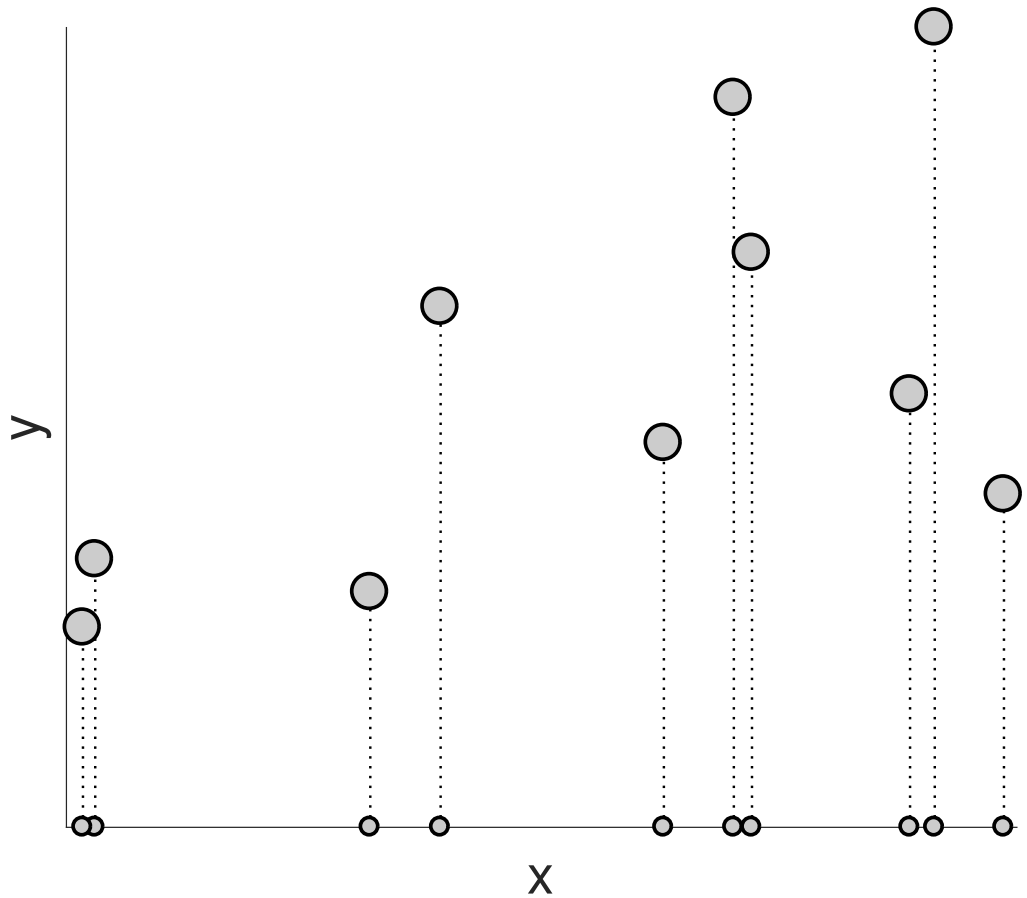
Subsampling for Ridge Regression via Regularized Volume Sampling

Michał Dereziński and Manfred Warmuth

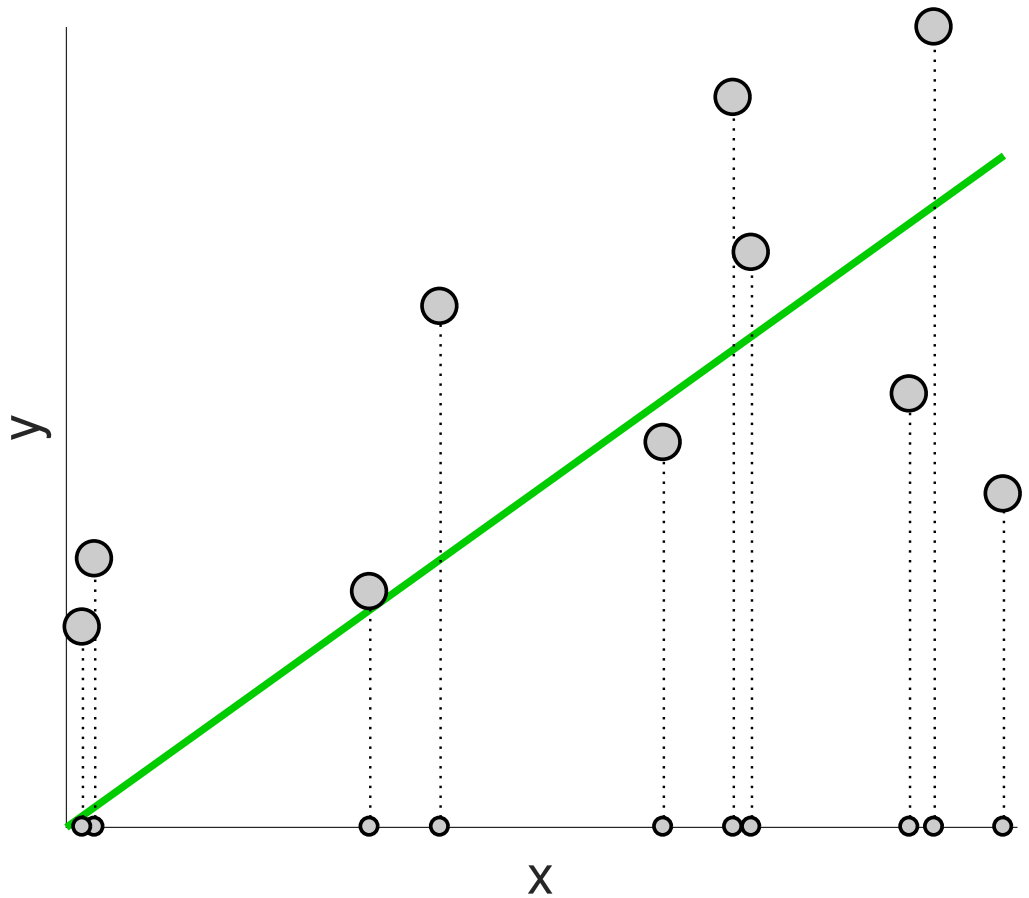
University of California at Santa Cruz

AISTATS'18, 4-9-18

Linear regression

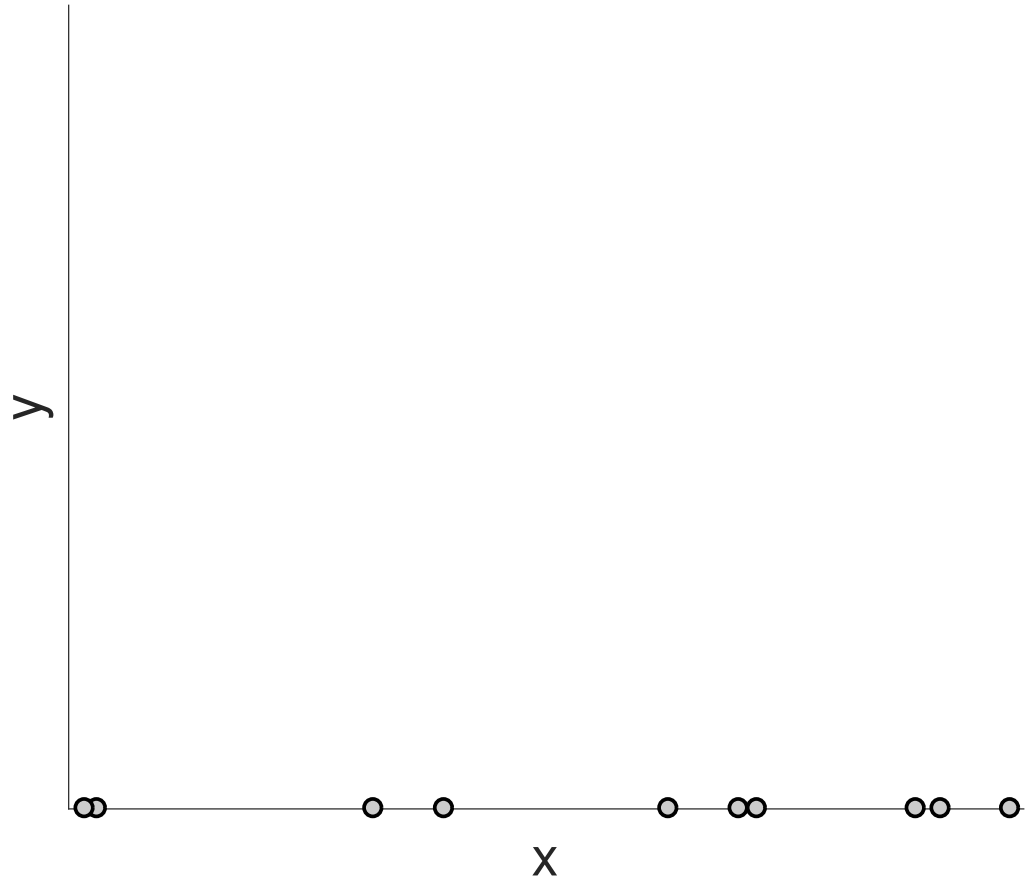


Optimal solution



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

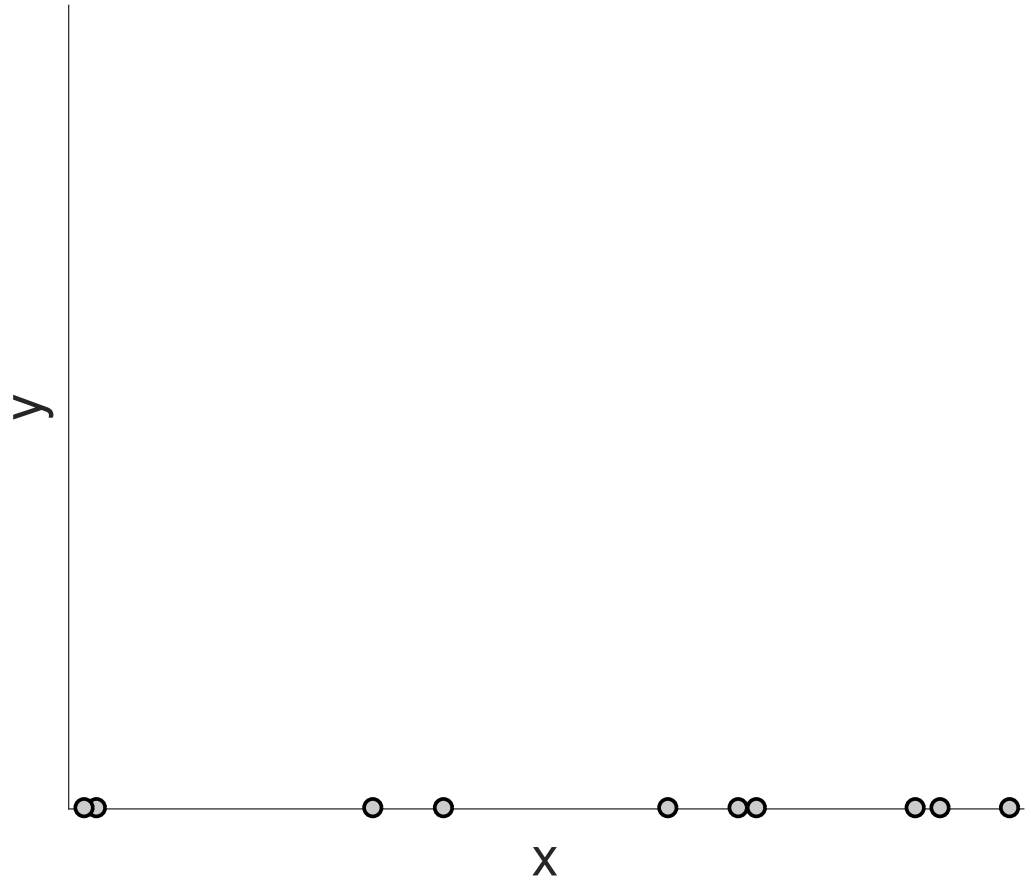
How many labels needed to get close to optimum?



- All x_i given
- But labels y_i unknown

Guess how many needed?

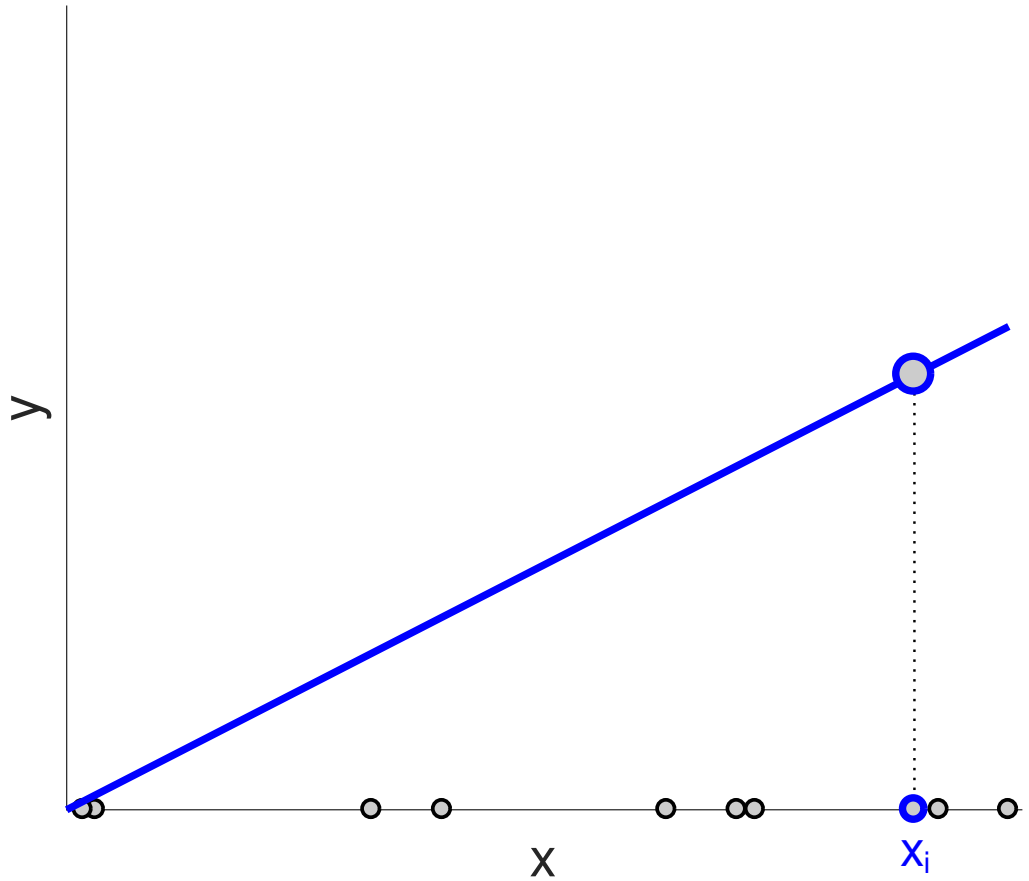
How many labels needed to get close to optimum?



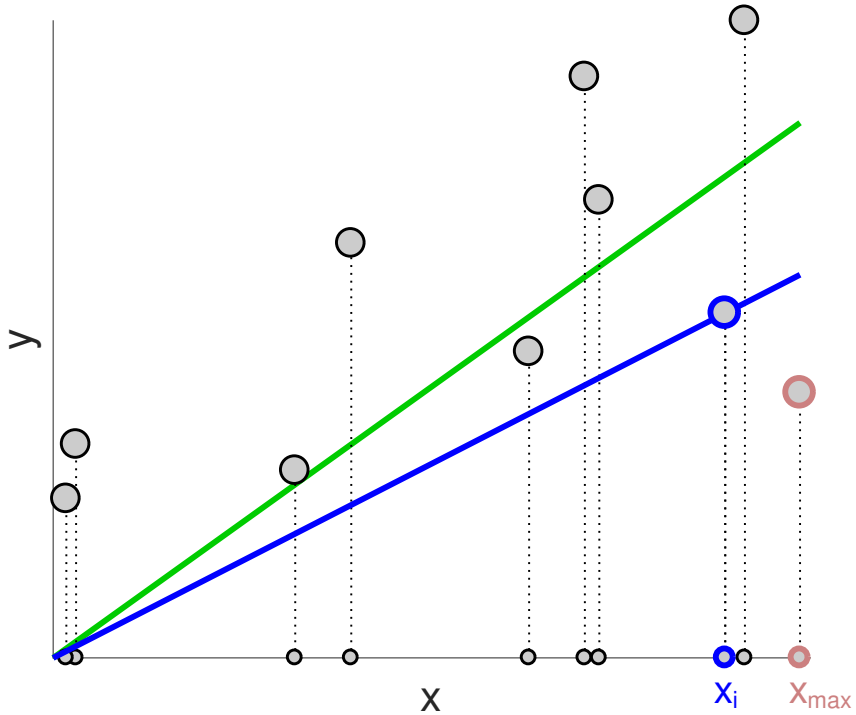
- All x_i given
- But labels y_i unknown

Guess how many needed?

Answer: one label



Which one?



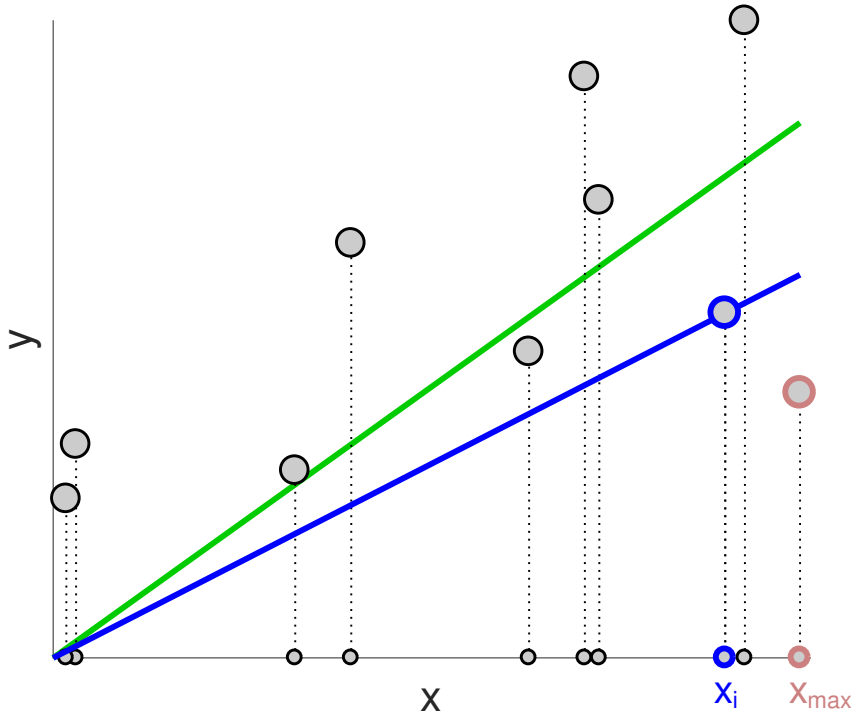
- x_{max} (furthest from 0) is bad
- any deterministic choice is bad

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

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

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

Which one?



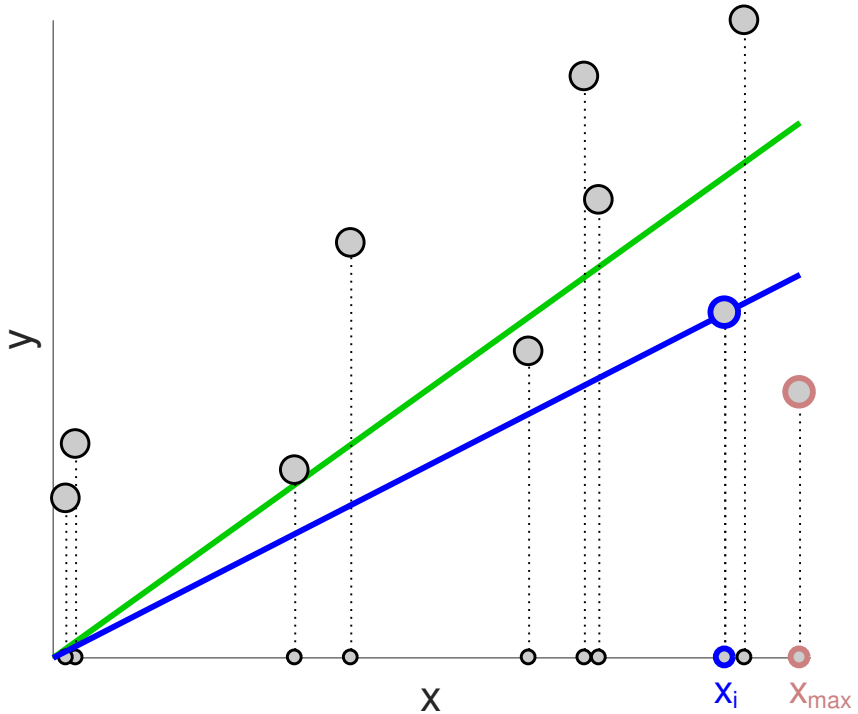
- x_{max} (furthest from 0) is bad
- any deterministic choice is bad

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

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

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

Which one?



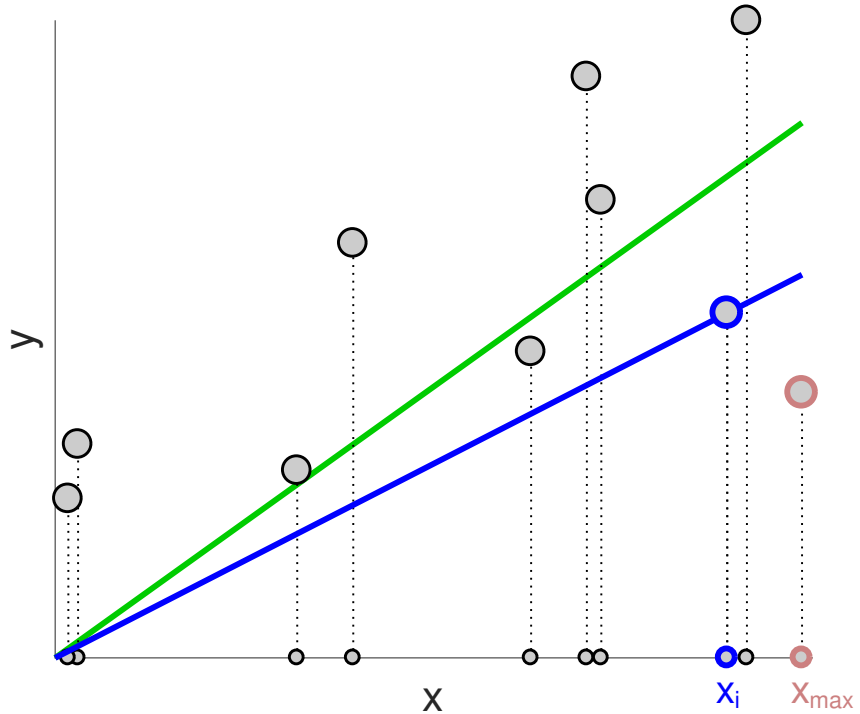
- x_{max} (furthest from 0) is bad
- any deterministic choice is bad

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

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

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

Which one?



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

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

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

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

Generalization: Volume Sampling

- Given n points $\mathbf{x}_i \in \mathbb{R}^d$
- Choose subset S of d indices
- Get labels $y_i \in \mathbb{R}$ for $i \in S$
- Find $\mathbf{w}^*(S)$: solution for the d examples $\{(\mathbf{x}_i, y_i) : i \in S\}$

Key prior result [DW17][1]: for any $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)$,

$$\mathbb{E}_S \sum_j (\mathbf{x}_j^\top \mathbf{w}^*(S) - \mathbf{x}_j^\top \mathbf{w}^*)^2 = \underline{d} \sum_j (\mathbf{x}_j^\top \mathbf{w}^* - y_j)^2$$

when S chosen $\sim$
squared volume of parallelopiped
spanned by the $\{\mathbf{x}_i : i \in S\}$

[1] DW. NIPS'17

Generalization: Volume Sampling

- Given n points $\mathbf{x}_i \in \mathbb{R}^d$
- Choose subset S of d indices
- Get labels $y_i \in \mathbb{R}$ for $i \in S$
- Find $\mathbf{w}^*(S)$: solution for the d examples $\{(\mathbf{x}_i, y_i) : i \in S\}$

Key prior result [DW17][1]: for any $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)$,

$$\mathbb{E}_S \sum_j (\mathbf{x}_j^\top \mathbf{w}^*(S) - \mathbf{x}_j^\top \mathbf{w}^*)^2 = \underline{d} \sum_j (\mathbf{x}_j^\top \mathbf{w}^* - y_j)^2$$

when S chosen $\sim$
squared volume of parallelopiped spanned by the $\{\mathbf{x}_i : i \in S\}$

Unbiasedness:

$$\mathbb{E}_S \mathbf{w}^*(S) = \mathbf{w}^*$$

[1] DW. NIPS'17

Generalization: Volume Sampling

- Given n points $\mathbf{x}_i \in \mathbb{R}^d$
- Choose subset S of d indices
- Get labels $y_i \in \mathbb{R}$ for $i \in S$
- Find $\mathbf{w}^*(S)$: solution for the d examples $\{(\mathbf{x}_i, y_i) : i \in S\}$

Key prior result [DW17][1]: for any $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)$,

$$\mathbb{E}_S \sum_j (\mathbf{x}_j^\top \mathbf{w}^*(S) - \mathbf{x}_j^\top \mathbf{w}^*)^2 = \underline{d} \sum_j (\mathbf{x}_j^\top \mathbf{w}^* - y_j)^2$$

when S chosen $\sim$
squared volume of parallelopiped
spanned by the $\{\mathbf{x}_i : i \in S\}$

Open:

Replace factor $\underline{d}$ with $\underline{\epsilon}$,
using small $|S| \geq d$

[1] DW. NIPS'17

Two types of volume sampling

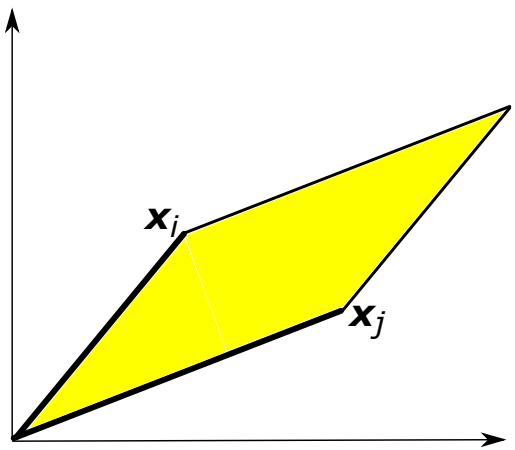
Let $\mathbf{X} \in \mathbb{R}^{d \times n}$ be full rank, $n \gg d$

$$\mathbf{X}_S = \begin{pmatrix} | & | \\ \mathbf{x}_i & \mathbf{x}_j \\ | & | \end{pmatrix}$$

$\det(\mathbf{X}_S \mathbf{X}_S^\top) =$ **squared** volume of parallelepiped $\mathcal{P}(\mathbf{x}_i, \mathbf{x}_j)$

Distribution over all s-element subsets S:

$$P(S) \sim \begin{cases} \det(\mathbf{X}_S^\top \mathbf{X}_S) & \text{if } s \leq d \text{ [DRVW06]}^1 \\ \det(\mathbf{X}_S \mathbf{X}_S^\top) & \text{if } s \geq d \text{ [AB13]}^2 \end{cases}$$



[1] [DRVW06] Deshpande et al. SODA'06

[2] [AB13] Avron and Boutsidis. JMAA'13

Prior work: efficient volume sampling algorithms

Sampling sets of size $s \leq d$:

1. Deshpande and Rademacher (FOCS'10) $O(snd^{\omega} \log d)$
2. Guruswami and Sinop (SODA'12) $O(snd^2)$

Sampling sets of size $s \geq d$:

1. Li et al. (NIPS'17) (approximate MCMC) $\widetilde{O}(s^3 nd^2)$
2. DW (NIPS'17) $O(n^2 d)$

We give an $O(nd^2)$ algorithm for $s \geq d$

Prior work: efficient volume sampling algorithms

Sampling sets of size $s \leq d$:

1. Deshpande and Rademacher (FOCS'10) $O(snd^{\omega} \log d)$
2. Guruswami and Sinop (SODA'12) $O(snd^2)$

Sampling sets of size $s \geq d$:

1. Li et al. (NIPS'17) (approximate MCMC) $\widetilde{O}(s^3nd^2)$
2. DW (NIPS'17) $O(n^2d)$

We give an $O(nd^2)$ algorithm for $s \geq d$

Our contributions: subsampling for linear regression

1. **Statistical**
   - 1.1 *Dimension-free error bounds* (random noise assumptions)

$$\text{Error} \;=\; O\left(\frac{\text{degrees of freedom}}{\text{subset size } s}\right)$$

   - 1.2 *Lower bounds*
     - volume sampling achieves near-optimal error bounds
     - no i.i.d. sampling works as well for small sample size s

2. **Computational**
   - 2.1 *Regularized volume sampling*
   - 2.2 *Faster sampling algorithm* $O(nd^2)$ for any $s \geq d$

Our contributions: subsampling for linear regression

1. **Statistical**
   - 1.1 *Dimension-free error bounds* (random noise assumptions)

$$\text{Error} \;=\; O\left(\frac{\text{degrees of freedom}}{\text{subset size } s}\right)$$

   - 1.2 *Lower bounds*
     - volume sampling achieves near-optimal error bounds
     - no i.i.d. sampling works as well for small sample size s

2. **Computational**
   - 2.1 *Regularized volume sampling*
   - 2.2 *Faster sampling algorithm* $O(nd^2)$ for any $s \geq d$

Our contributions: subsampling for linear regression

1. **Statistical**
   - 1.1 *Dimension-free error bounds* (random noise assumptions)

$$\text{Error} \;=\; O\left(\frac{\text{degrees of freedom}}{\text{subset size } s}\right)$$

   - 1.2 *Lower bounds*
     - volume sampling achieves near-optimal error bounds
     - no i.i.d. sampling works as well for small sample size s

2. **Computational**
   - 2.1 *Regularized volume sampling*
   - 2.2 *Faster sampling algorithm* $O(nd^2)$ for any $s \geq d$

Statistical model: linear labels plus bounded noise

Fixed design matrix: $\mathbf{X} \in \mathbb{R}^{d \times n}$, linear model: $\mathbf{w}^* \in \mathbb{R}^d$

$$\mathbf{y} = \mathbf{X}^\top \mathbf{w}^* + \boldsymbol{\xi}, \quad \text{where} \quad \mathbb{E}[\boldsymbol{\xi}] = 0, \quad \text{and} \quad \mathrm{Var}[\boldsymbol{\xi}] \preceq \sigma^2 \mathbf{I}$$

Goal: Minimize *Mean Squared Prediction Error* using few labels

$$\mathrm{MSPE}(\widehat{\mathbf{w}}) = \mathbb{E}\, \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i^\top \widehat{\mathbf{w}} - \mathbf{x}_i^\top \mathbf{w}^*)^2$$

Statistical model: linear labels plus bounded noise

Fixed design matrix: $\mathbf{X} \in \mathbb{R}^{d \times n}$, linear model: $\mathbf{w}^* \in \mathbb{R}^d$

$$\mathbf{y} = \mathbf{X}^\top \mathbf{w}^* + \boldsymbol{\xi}, \quad \text{where} \quad \mathbb{E}[\boldsymbol{\xi}] = 0, \quad \text{and} \quad \mathrm{Var}[\boldsymbol{\xi}] \preceq \sigma^2 \mathbf{I}$$

Goal: Minimize *Mean Squared Prediction Error* using few labels

$$\mathrm{MSPE}(\widehat{\mathbf{w}}) = \mathbb{E}\, \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i^\top \widehat{\mathbf{w}} - \mathbf{x}_i^\top \mathbf{w}^*)^2$$

Dimension-free error bound

Ridge estimator: $\widehat{\mathbf{w}}^*_\lambda(S) = \overbrace{\underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{X}_S^\top \mathbf{w} - \mathbf{y}_S\|^2 + \lambda\|\mathbf{w}\|^2}^{(\mathbf{X}_S\mathbf{X}_S^\top + \lambda \mathbf{I})^{-1}\mathbf{X}_S\mathbf{y}_S}$

Main theorem

If $\lambda \leq \frac{\sigma^2}{\|\mathbf{w}^*\|^2}$, then there is a distribution over subsets S of size s

$$\text{s.t.} \quad \text{MSPE}(\widehat{\mathbf{w}}^*_\lambda(S)) \leq \frac{\sigma^2 d_\lambda}{s - d_\lambda + 1} = O\left(\frac{\sigma^2 d_\lambda}{s}\right),$$

where $d_\lambda = \sum_{i=1}^d \frac{\lambda_i}{\lambda_i + \lambda} < d, \quad \{\lambda_i\}$ - eigenvalues of $\mathbf{X}\mathbf{X}^\top$

d_λ - degrees of freedom (statistical dimension) of $\mathbf{X}$, given λ

Note: Under decaying eigenvalues, $d_\lambda \ll d$

Dimension-free error bound

Ridge estimator: $\widehat{\mathbf{w}}^*_\lambda(S) = \overbrace{\underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{X}_S^\top \mathbf{w} - \mathbf{y}_S\|^2 + \lambda\|\mathbf{w}\|^2}^{(\mathbf{X}_S\mathbf{X}_S^\top + \lambda\mathbf{I})^{-1}\mathbf{X}_S\mathbf{y}_S}$

Main theorem

If $\lambda \leq \frac{\sigma^2}{\|\mathbf{w}^*\|^2}$, then there is a distribution over subsets S of size s

$$\text{s.t.} \quad \text{MSPE}(\widehat{\mathbf{w}}^*_\lambda(S)) \leq \frac{\sigma^2 d_\lambda}{s - d_\lambda + 1} = O\left(\frac{\sigma^2 d_\lambda}{s}\right),$$

where $d_\lambda = \sum_{i=1}^d \frac{\lambda_i}{\lambda_i + \lambda} < d$, $\quad \{\lambda_i\}$ - eigenvalues of $\mathbf{X}\mathbf{X}^\top$

d_λ - degrees of freedom (statistical dimension) of $\mathbf{X}$, given λ

Note: Under decaying eigenvalues, $d_\lambda \ll d$

Regularized volume sampling

Naive idea:

$$\text{replace} \quad P(S) \sim \det(\mathbf{X}_S \mathbf{X}_S^\top),$$

$$\text{with} \quad P(S) \sim \det(\mathbf{X}_S \mathbf{X}_S^\top + \lambda \mathbf{I})$$

Problem: Not clear how to sample efficiently

Intuition: No simple extension of the Cauchy-Binet formula

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

$$\sum_{S:\ |S|=s} \det(\mathbf{X}_S \mathbf{X}_S^\top + \lambda \mathbf{I}) = ???$$

Regularized volume sampling

Naive idea:

$$\text{replace} \quad P(S) \sim \det(\mathbf{X}_S \mathbf{X}_S^\top),$$

$$\text{with} \quad P(S) \sim \det(\mathbf{X}_S \mathbf{X}_S^\top + \lambda \mathbf{I})$$

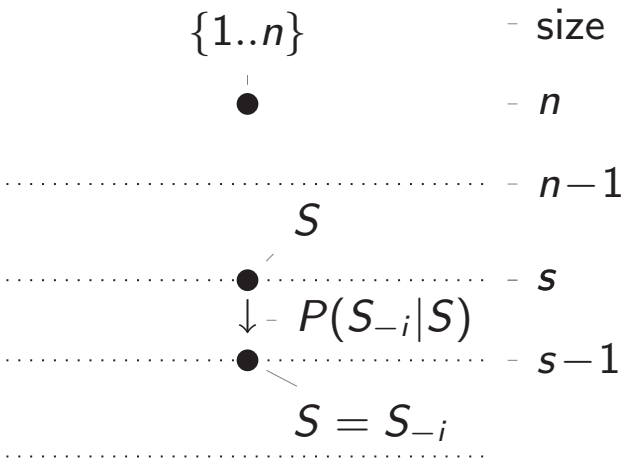
Problem: Not clear how to sample efficiently

Intuition: No simple extension of the Cauchy-Binet formula

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

$$\sum_{S:\ |S|=s} \det(\mathbf{X}_S \mathbf{X}_S^\top + \lambda \mathbf{I}) = ???$$

Reverse iterative sampling



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

Sample index $i \in S$

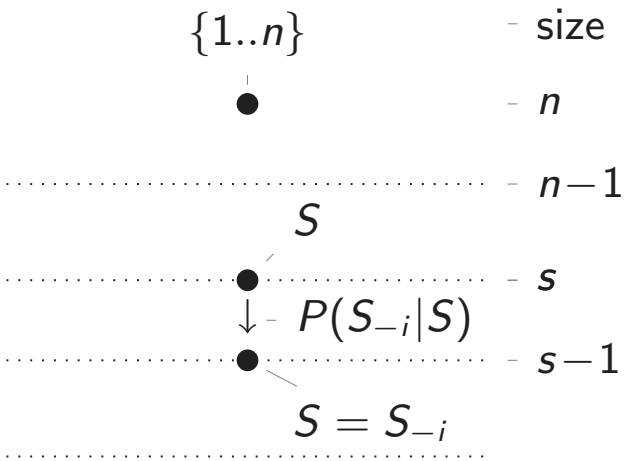
Go to set $S_{-i} = S - \{i\}$

Repeat until desired size

$$P(S_{-i}|S) \sim \frac{\det(\mathbf{X}_{S_{-i}}\mathbf{X}_{S_{-i}}^\top + \lambda\mathbf{I})}{\det(\mathbf{X}_S\mathbf{X}_S^\top + \lambda\mathbf{I})} = 1 - \mathbf{x}_i^\top(\mathbf{X}_S\mathbf{X}_S^\top + \lambda\mathbf{I})^{-1}\mathbf{x}_i$$

Note: not the same as $P(S) \sim \det(\mathbf{X}_S\mathbf{X}_S^\top + \lambda\mathbf{I})$ (unless $\lambda = 0$)

Reverse iterative sampling



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

Sample index $i \in S$

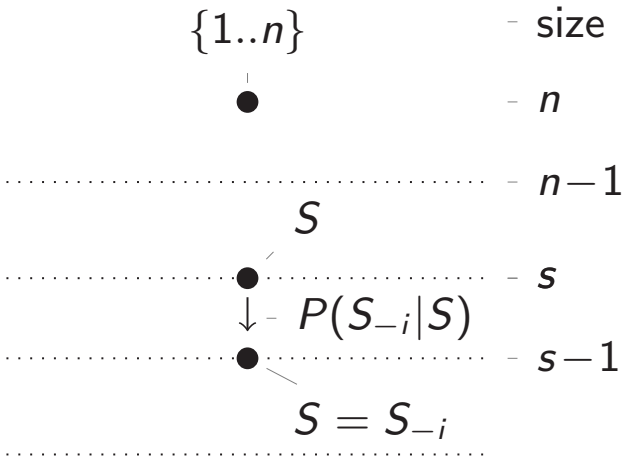
Go to set $S_{-i} = S - \{i\}$

Repeat until desired size

$$P(S_{-i}|S) \sim \frac{\det(\mathbf{X}_{S_{-i}}\mathbf{X}_{S_{-i}}^\top + \lambda\mathbf{I})}{\det(\mathbf{X}_S\mathbf{X}_S^\top + \lambda\mathbf{I})} = 1 - \mathbf{x}_i^\top(\mathbf{X}_S\mathbf{X}_S^\top + \lambda\mathbf{I})^{-1}\mathbf{x}_i$$

Note: not the same as $P(S) \sim \det(\mathbf{X}_S\mathbf{X}_S^\top + \lambda\mathbf{I})$ (unless $\lambda = 0$)

Reverse iterative sampling



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

Sample index $i \in S$

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

Repeat until desired size

$$P(S_{-i}|S) \sim \frac{\det(\mathbf{X}_{S_{-i}}\mathbf{X}_{S_{-i}}^\top + \lambda\mathbf{I})}{\det(\mathbf{X}_S\mathbf{X}_S^\top + \lambda\mathbf{I})} = 1 - \mathbf{x}_i^\top(\mathbf{X}_S\mathbf{X}_S^\top + \lambda\mathbf{I})^{-1}\mathbf{x}_i$$

Note: not the same as $P(S) \sim \det(\mathbf{X}_S\mathbf{X}_S^\top + \lambda\mathbf{I})$ (unless $\lambda = 0$)

Key Expectation Bound

Lemma

For λ-regularized volume sampling of set S of size s

$$\mathbb{E}_S \left(\mathbf{X}_S \mathbf{X}_S^\top + \lambda \mathbf{I}\right)^{-1} \preceq \frac{n - d_\lambda + 1}{s - d_\lambda + 1} (\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I})^{-1}$$

(for $\lambda = 0$, this was shown in [DW17])

Proof of main theorem

Bias-variance decomposition for fixed S:

$$\mathrm{MSPE}(\widehat{\mathbf{w}}_\lambda^*(S) \,|\, S) = (\mathrm{bias})^2 + \mathrm{variance} \leq \frac{\sigma^2}{n} \mathrm{tr}(\mathbf{X}^\top (\mathbf{X}_S \mathbf{X}_S^\top + \lambda \mathbf{I})^{-1} \mathbf{X})$$

Take expectation over S and apply the lemma.

Key Expectation Bound

Lemma

For λ-regularized volume sampling of set S of size s

$$\mathbb{E}_S \left(\mathbf{X}_S \mathbf{X}_S^\top + \lambda \mathbf{I}\right)^{-1} \preceq \frac{n - d_\lambda + 1}{s - d_\lambda + 1} (\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I})^{-1}$$

(for $\lambda = 0$, this was shown in [DW17])

Proof of main theorem

Bias-variance decomposition for fixed S:

$$\text{MSPE}(\widehat{\mathbf{w}}_\lambda^*(S) \,|\, S) = (\text{bias})^2 + \text{variance} \leq \frac{\sigma^2}{n} \text{tr}(\mathbf{X}^\top (\mathbf{X}_S \mathbf{X}_S^\top + \lambda \mathbf{I})^{-1} \mathbf{X})$$

Take expectation over S and apply the lemma.

Volume sampling vs leverage score sampling

Leverage score sampling:
examples selected i.i.d. w.p. $\sim \mathbf{x}_i^\top (\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{x}_i$.

How many labels needed for MSPE($\widehat{\mathbf{w}}$) $\leq \sigma^2$? (σ^2 - label noise)

1. *Volume sampling* $2d_\lambda$ labels
2. *Any i.i.d. sampling* $\Omega(d_\lambda \ln d_\lambda)$ labels (lower bound)

Our volume sampling is as fast as computing exact leverage scores

Volume sampling vs leverage score sampling

Leverage score sampling:
examples selected i.i.d. w.p. $\sim \mathbf{x}_i^\top (\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{x}_i$.

How many labels needed for $\mathrm{MSPE}(\widehat{\mathbf{w}}) \leq \sigma^2$? ($\sigma^2$ - label noise)

1. *Volume sampling* $2d_\lambda$ labels
2. *Any i.i.d. sampling* $\Omega(d_\lambda \ln d_\lambda)$ labels (lower bound)

Our volume sampling is as fast as computing exact leverage scores

Volume sampling vs leverage score sampling

Leverage score sampling:
examples selected i.i.d. w.p. $\sim \mathbf{x}_i^\top (\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{x}_i$.

How many labels needed for $\mathrm{MSPE}(\widehat{\mathbf{w}}) \leq \sigma^2$? ($\sigma^2$ - label noise)

1. *Volume sampling* $2d_\lambda$ labels
2. *Any i.i.d. sampling* $\Omega(d_\lambda \ln d_\lambda)$ labels (lower bound)

Our volume sampling is as fast as computing exact leverage scores

Volume sampling vs leverage score sampling

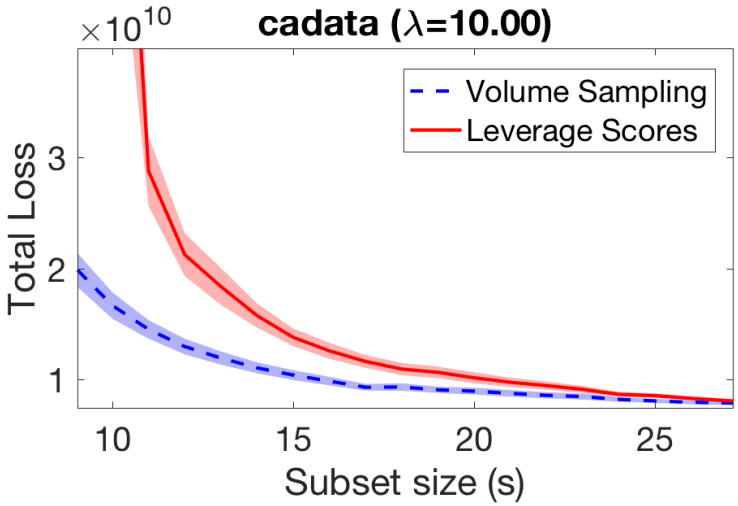
Leverage score sampling:
examples selected i.i.d. w.p. $\sim \mathbf{x}_i^\top (\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{x}_i$.

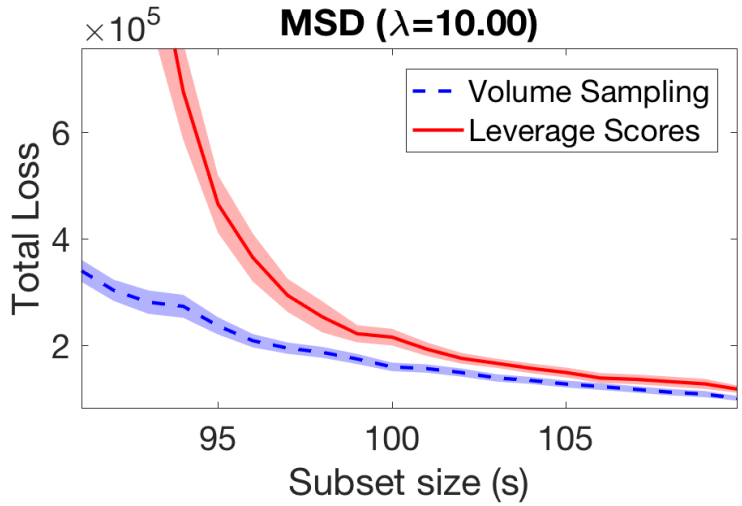
How many labels needed for MSPE($\widehat{\mathbf{w}}$) $\leq \sigma^2$? (σ^2 - label noise)

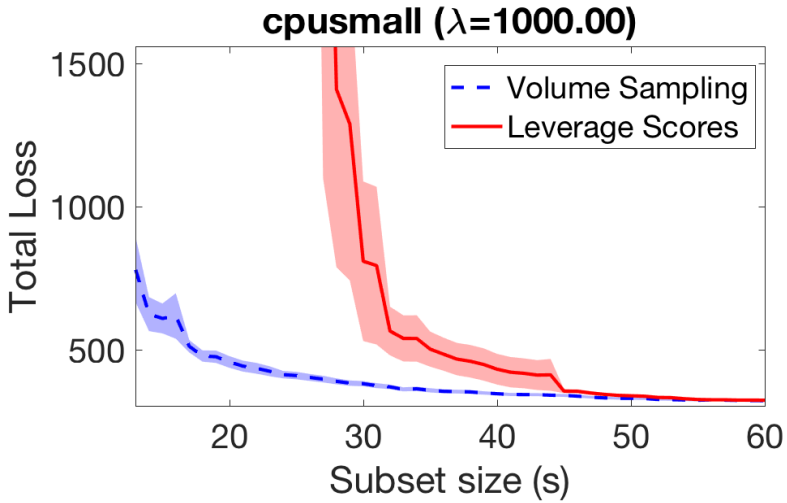
1. *Volume sampling* $2d_\lambda$ labels
2. *Any i.i.d. sampling* $\Omega(d_\lambda \ln d_\lambda)$ labels (lower bound)

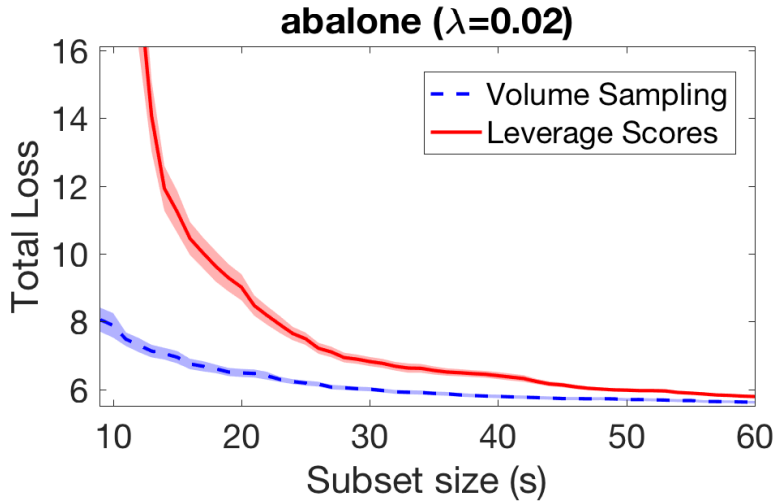
Our volume sampling is as fast as computing exact leverage scores

Volume sampling vs leverage scores on real data

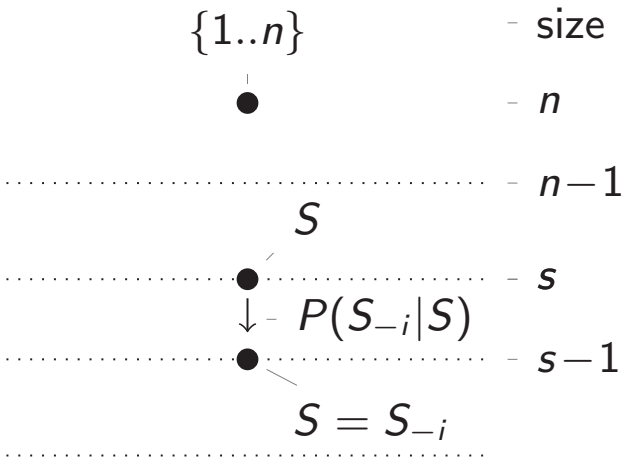








Simple algorithm for volume sampling



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

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

1. Sample i uniformly from set S,
2. Compute $h_i = 1 - \mathbf{x}_i^\top (\mathbf{X}_S \mathbf{X}_S^\top + \lambda \mathbf{I})^{-1} \mathbf{x}_i$,
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: $P(S_{-i}|S) \sim 1 - \mathbf{x}_i^\top (\mathbf{X}_S \mathbf{X}_S^\top + \lambda \mathbf{I})^{-1} \mathbf{x}_i$

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

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

} one trial

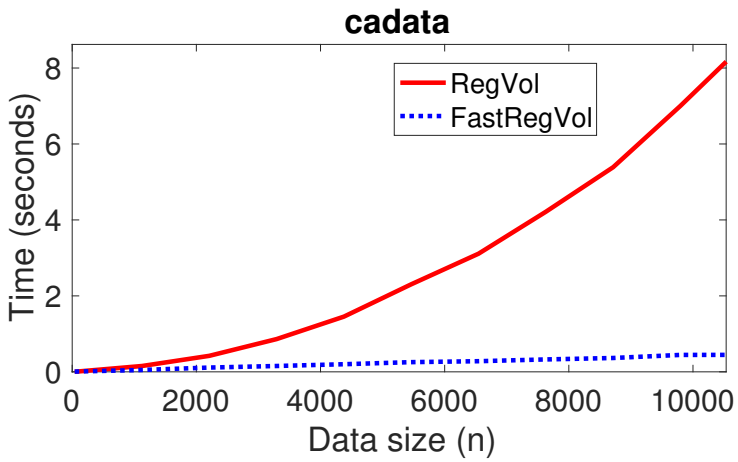
We show: Number of trials per step is constant w.h.p.

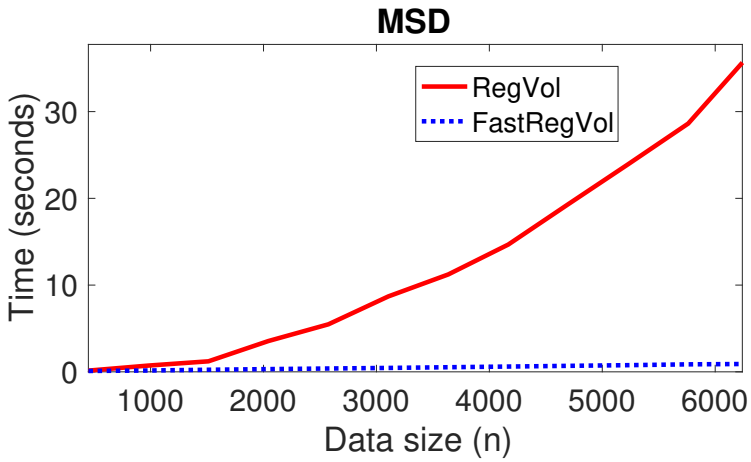
Runtime: $\overbrace{\text{steps}}^{n-s} \times \overbrace{\text{trials per step}}^{O(1)} \times \overbrace{\text{compute } h_i}^{O(d^2)} = O(nd^2)$

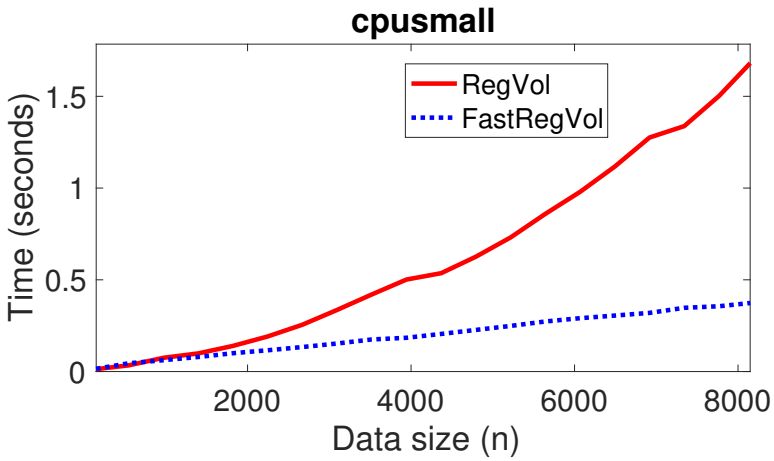
Result: Linear dependence on n

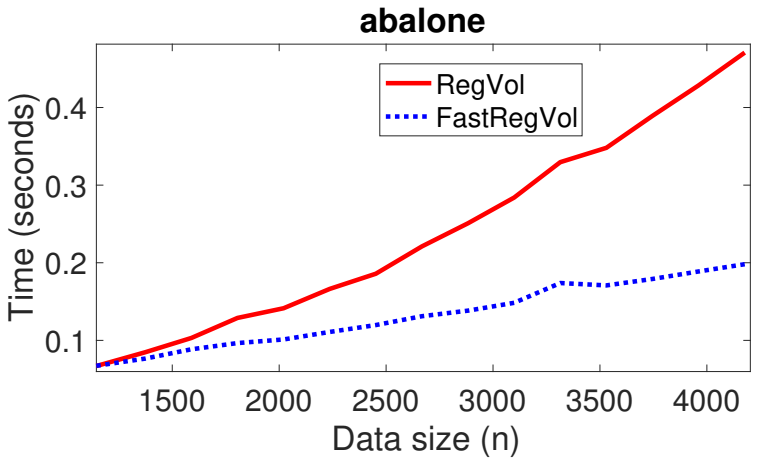
Faster volume sampling on real data

RegVol: simple algorithm *FastRegVol:* rejection sampling









Conclusions

1. Dimension-free error bounds for subsampled regression

2. Optimal subsampling must be joint

3. Why pick volume sampling over leverage score sampling?
   - 3.1 Better error bounds for small sample sizes,
   - 3.2 Nearly the same computational efficiency.

Conclusions

1. Dimension-free error bounds for subsampled regression

2. Optimal subsampling must be joint

3. Why pick volume sampling over leverage score sampling?
   - 3.1 Better error bounds for small sample sizes,
   - 3.2 Nearly the same computational efficiency.

Conclusions

1. Dimension-free error bounds for subsampled regression

2. Optimal subsampling must be joint

3. Why pick volume sampling over leverage score sampling?
   - 3.1 Better error bounds for small sample sizes,
   - 3.2 Nearly the same computational efficiency.

Thank you