New Subsampling Algorithms for Fast Least Squares Regression

Paramveer S. Dhillon[1], Yichao Lu[2], Dean Foster[2] and Lyle Ungar[1]
{[1]CIS,[2]Statistics (Wharton)} University of Pennsylvania, Philadelphia, PA, U.S.A

Background

- Problem: Estimation of ordinary least squares (OLS) regression when $n \gg p$ (n observations, p features).
- OLS Regression: $Y = \mathbf{X}w_0 + \epsilon$, $\epsilon \sim \mathcal{N}(0,1)$.
- MLE solution $\longrightarrow \hat{w} = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top Y$.
  - \+ Running Time (FLOPS): $O(np^2)$.
  - \+ Error bound: $\|w_0 - \hat{w}\| \to O(\sqrt{\frac{p}{n}})$
- Current state of the art: Preconditioning based approaches [(Drineas)+ 07, (Rokhlin)+ 08].
  1. Transform the data with randomized Hadamard (SRHT) or Fast Fourier Transform (FFT).
  2. Uniformly subsample the resulting matrix ($n_{subs} = O(p)$).
  3. Estimate the OLS on this smaller matrix.
     - \+ Running Time (FLOPS): $O(\max(np\log p,\ n_{subs}p^2))$.
     - \+ Error bound: $\|w_0 - \hat{w}\| \to O(\sqrt{\frac{p}{n_{subs}}})$.

The Algorithms

- Either precondition the data matrix $\mathbf{X}$ and then subsample (*fixed design*) or directly subsample (*sub-gaussian random design*) (If you believe the data is i.i.d.).

1. **Full Subsampling Algorithm (FS)** Subsample $\mathbf{X}$ and Y, then $\hat{w}_{FS} = (\mathbf{X_{subs}}^\top \mathbf{X_{subs}})^{-1}\mathbf{X_{subs}}^\top Y_{subs}$.
   - \+ Similar to [(Drineas)+ 07], but novel error analysis.
2. **Covariance Subsampling Algorithm (CovS)** $\longrightarrow \hat{w}_{CovS} = (\mathbf{X_{subs}}^\top \mathbf{X_{subs}})^{-1}\mathbf{X}^\top Y$.
3. **Uluru** $\longrightarrow$ Two stage algorithm
   (a) Stage 1: Use **FS** to estimate $\hat{w}_{FS}$.
   (b) Stage 2: Use **CovS** to estimate $\hat{w}_{correct}$ on the remaining observations ($n_{rem} = n \backslash n_{subs}$).
   (c) Perform Sampling Correction: $\widehat{w}_{Uluru} = \widehat{w}_{FS} + \widehat{w}_{correct}$.

Uluru

Methods	Running Time O(FLOPS)	Error bound
OLS	$O(n\,p^2)$	$O(\sqrt{p/n})$
FS	$O(nr\,p^2)$	$O(\sqrt{p/nr})$
CovS	$O(nr\,p^2 + n\,p)$	*
Uluru	$O(nr\,p^2 + n\,p)$	$O(\sqrt{p/n})$

- We do not increase the error of **Uluru** by using less data in estimating the covariance matrix. So our estimate of the *quadratic* term is as solid as the rock formation **Uluru**!

Theory (Summary)

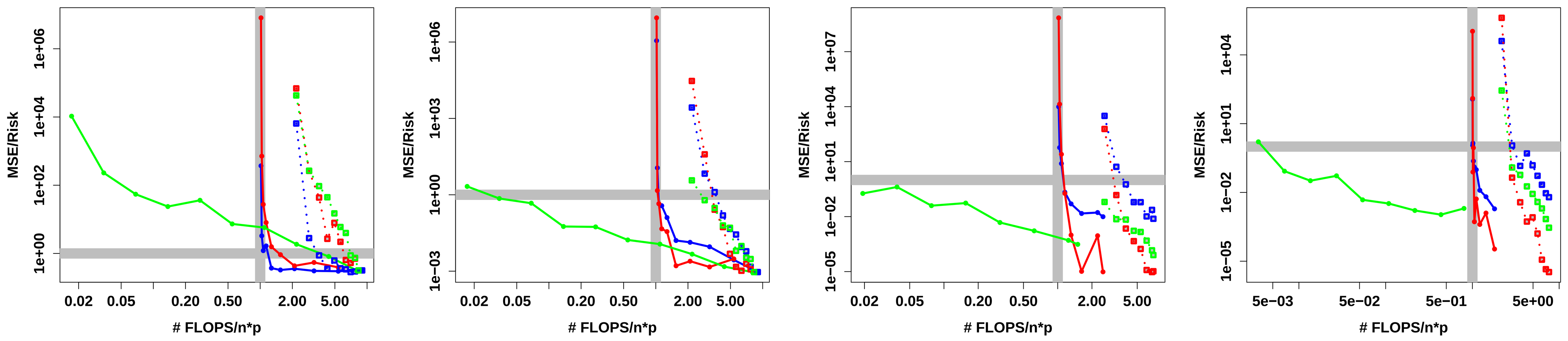
- When $n_{subs} \ll n_{rem}$, keeping only the dominating terms, the results can be summarized as: With failure probability less than some fixed number, the algorithms have the following error bounds.

1. **FS** $\longrightarrow O(\sigma\sqrt{\frac{p}{n_{subs}}})$.
2. **CovS** $\longrightarrow O(\sqrt{\frac{p}{n_{subs}}}\|w\| + \sigma\sqrt{\frac{p}{n}})$.
3. **Uluru** $\longrightarrow O(\sigma\frac{p}{n_{subs}} + \sigma\sqrt{\frac{p}{n}})$.

- If the second term for the error of the **Uluru** algorithm dominates, i.e. if $r(=\frac{n_{subs}}{n}) > O(\sqrt{p/n})$ then the error bound of **Uluru** $\approx O(\sigma\sqrt{\frac{p}{n}})$ (completely independent of r!).
- The threshold for r only depends on the properties of design matrix (n, p) and not on the noise level σ.
  - **FS** and **CovS** do not have this property.

Experiments

- Results for synthetic datasets (Plots 1-2, low signal and high signal) and for real world datasets (Plots 3-4, CPUSMALL, CADATA). Color scheme: + (Green)-FS, + (Blue)-CovS, + (Red)-Uluru. The solid lines indicate no preconditioning (i.e. random design) and dashed lines indicate fixed design with Randomized Hadamard preconditioning. The FLOPS reported are the theoretical values.



Conclusion

Uluru has a runtime of $O(np)$ and obtains error bound of $O(\sqrt{\frac{p}{n}})$ which is the same as full OLS and is independent of amount of subsampling.