Householder Sketch for Accurate and Accelerated Least-Mean-Squares Solvers

ICML 2021

Jyotikrishna Dass Rabi Mahapatra

{dass.jyotikrishna,rabi}@tamu.edu

Department of Computer Science and Engineering

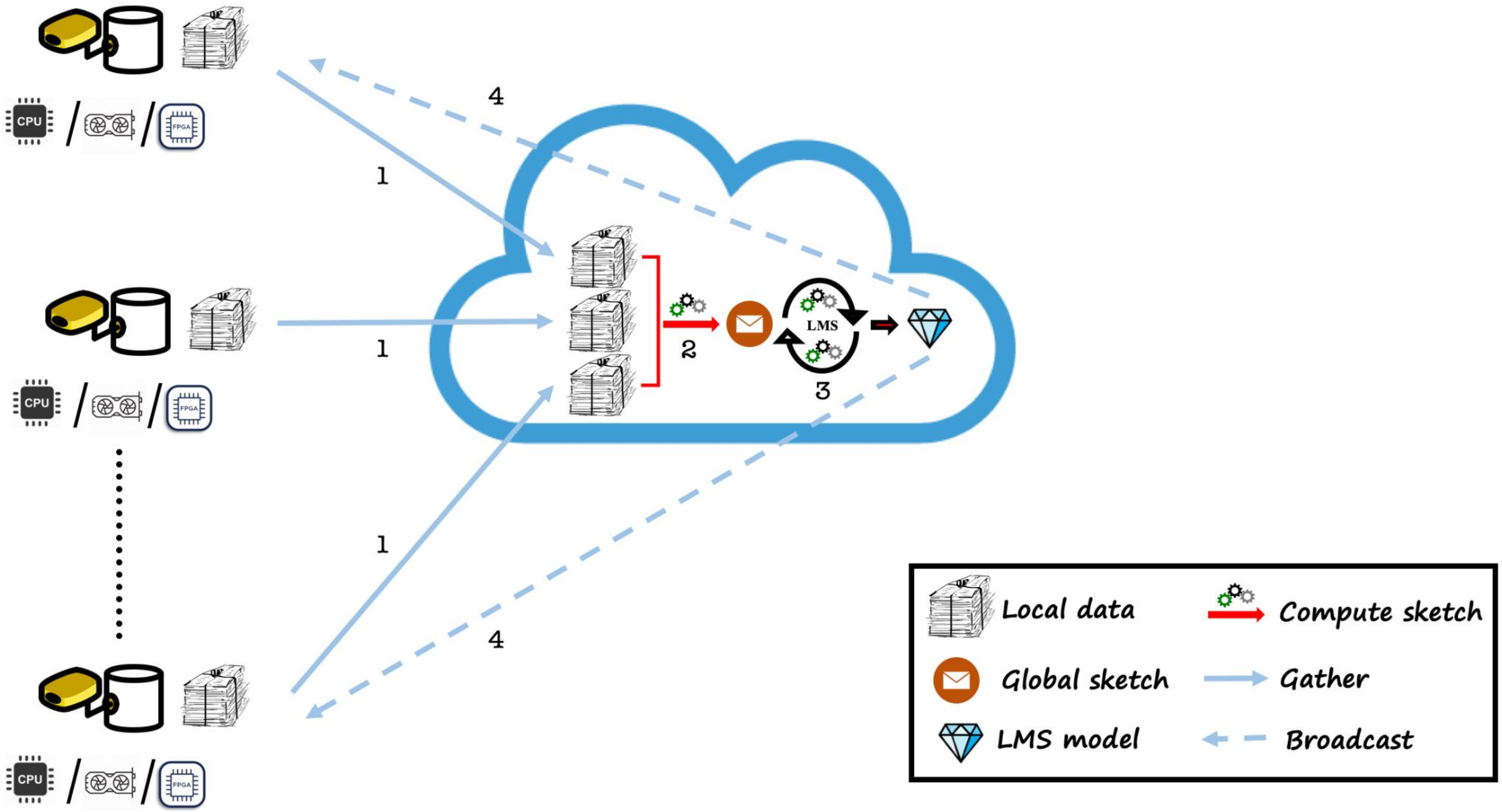
TEXAS A&M UNIVERSITY

Sketching

A compressed mapping of few or all data points (***X***) in a data set to generate **data summary** called ***Sketch (S)*** to preserve or approximate the covariance matrix, i.e.,

$$S^T S \cong X^T X$$

Sketch-based ML Framework



Least-Mean-Squares (LMS)

$$\min_{\mathbf{w}} f(\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2) + g(\mathbf{w}).$$

LINEAR REGRESSION, $f(z) = z^2$, and $g(\mathbf{w}) = 0$.

$$(\mathbf{X}^T\mathbf{X})\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

RIDGE REGRESSION, $f(z) = z^2$, and $g(\mathbf{w}) = \lambda\|\mathbf{w}\|_2$, where, $\lambda > 0$,

$$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

Our focus is on theoretically accurate summary of input data which could be directly plugged to accelerate scikit-learn LMS solvers

Inspiration

(Maalouf et al.)[1] proposed **LMS-BOOST**

- *Coreset-Sketch fusion* algorithm
- Faster implementation of Caratheodory Theorem (1907)
- *Accurately solve* and *accelerate* LMS solvers in scikit-learn library **upto 100x**
  - summarizes input data $\mathcal{X}$ into matrix $\boldsymbol{S}$ of size $O(d^2) \times d$
  - preserves the input covariance, i.e. $\boldsymbol{S}^T\boldsymbol{S} = \mathcal{X}^T\mathcal{X}$
  - computational time complexity of $O(nd^2 + \log(n) \times d^8)$

Claim 1: QR decomposition is relatively time-consuming.

Claim 2: QR decomposition is unsuitable for exact factorization for streaming data.

[1] Maalouf, A., Jubran, I., and Feldman, D. "*Fast and accurate least-mean-squares solvers*". in Advances in Neural Information Processing Systems, pp. 8305–8316, 2019

Contributions

Test and Check validity of the above claims made against the QR decomposition as a candidate for data summary via extensive theoretical and empirical analysis

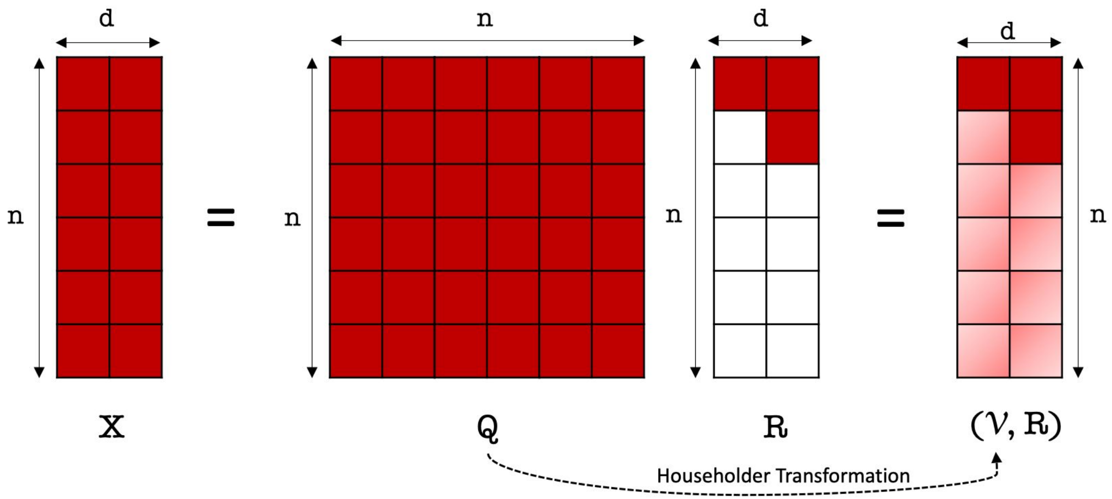
Q1: Whether a classical and simple approach such as QR decomposition could (theoretically) accurately solve and accelerate common LMS solvers compared to the above state of the art recursive and clustering-based fusion algorithm?

Q2: Whether a numerically stable algorithm could generate accurate distributed sketches via exact factorization on streaming data?

Householder-QR

Theorem 2.1 (Householder-QR (Golub & Van Loan, 2012)). *Let matrix $X \in \mathbb{R}^{n \times d}$ with $n > d$. Householder QR decomposition of X generates set of d Householder matrices $\mathcal{H}$ and an $n \times d$ upper trapezoidal matrix R. The Householder matrices are stored as a set of d Householder reflectors $\mathcal{V}$. Total memory footprint of above factors is nd elements with time complexity of $O(nd^2)$ for $n \gg d$.*

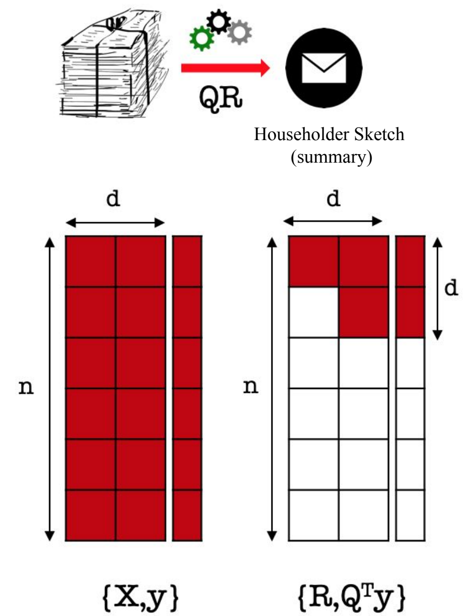
$$X = QR \text{ , where, } Q^TQ = QQ^T = I$$



memory consumption	$O(nd)$	$O(n^2)$	$O(d^2)$	$O(nd)$

Householder Sketch

Theorem 2.2 (Householder Sketch). *Let $X \in \mathbb{R}^{n\times d}$ be the original data matrix, $y \in \mathbb{R}^n$ be the corresponding output label or response vector, and $n \gg d$. Let $X = QR$ be Householder QR decomposition. Then, $(R, Q^T y)$ is a memory-efficient and theoretically accurate sketch of original data (X, y) such that $X^T X = R^T R$, and has memory footprint of $\left(\frac{d(d+3)}{2}\right)$ elements, computed in time $O(nd^2)$.*



memory consumption $O(nd)$ $O(d^2)$

Householder Sketch for LMS

Least-Mean-Squares

$$\min_{\mathbf{w}} f(\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2) + g(\mathbf{w}).$$

$$\min_{\mathbf{w}} f(\|\mathbf{Q}\mathbf{R}\mathbf{w} - \mathbf{y}\|_2) + g(\mathbf{w}).$$

$$\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2 = \|\mathbf{Q}\mathbf{R}\mathbf{w} - \mathbf{y}\|_2 = \|\mathbf{Q}\mathbf{R}\mathbf{w} - \mathbf{Q}\mathbf{Q}^T\mathbf{y}\|_2 = \|\mathbf{Q}\|_2 \, \|\mathbf{R}\mathbf{w} - \mathbf{Q}^T\mathbf{y}\|_2 = \|\mathbf{R}\mathbf{w} - \mathbf{Q}^T\mathbf{y}\|_2$$

(LMS) Accurate Sketch (LMS-QR)

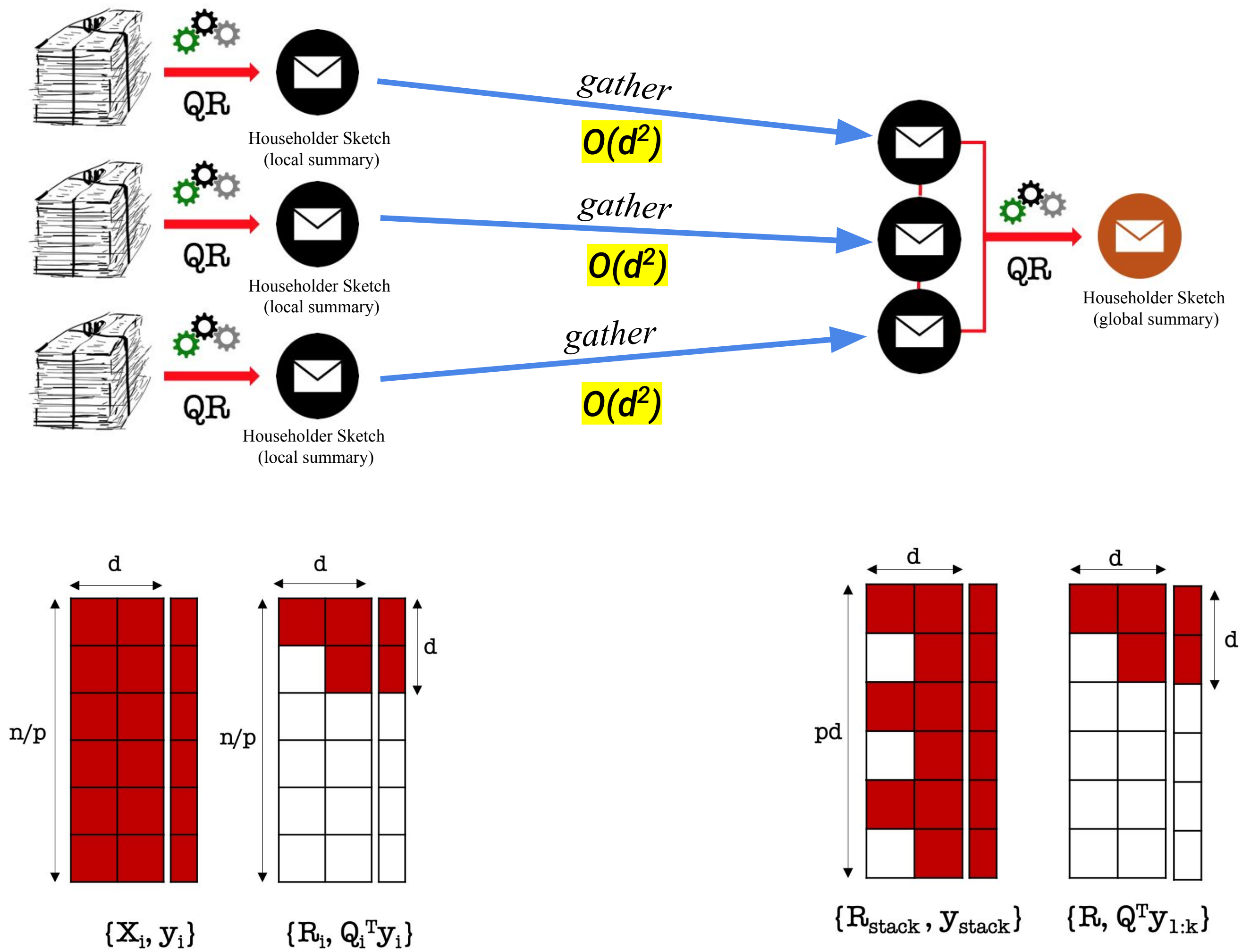
$$R^T R = X^T X$$

Distributed Householder Sketches

Theorem 3.1 (Distributed Householder-QR (Dass et al., 2018)). *Let $X = (X_1^T|\dots|X_p^T)^T$, where, $X_i \in \mathbb{R}^{\hat{n}\times d}$ be local data matrix of parallel worker, $i = 1,\dots,p$, where $\hat{n} \gg d$, and, $n = p\hat{n}$. Let, $X_i = Q_iR_i$ be constructed via local* Householder-QR *(see Algorithm 1) for each $i = 1,\dots,p$, in parallel. Then, $X = QR$ for the complete data matrix can be constructed exactly, such that $Q = \mathtt{diag}(Q_1,\dots,Q_p)Q_M$, and $R = R_M$, where $R_{stack} = Q_MR_M$ via another* Householder-QR *on $R_{stack} = (R_1^T|\dots|R_p^T)^T$ gathered from all workers. The above* Distributed Householder-QR *has a computational time complexity of $O(\frac{n}{p}d^2)$, with a communicated data volume of $(\frac{d(d+1)}{2})$ elements by each worker.*

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ . \\ . \\ \mathbf{X}_p \end{pmatrix} = \begin{pmatrix} \mathbf{Q}_1\mathbf{R}_1 \\ \mathbf{Q}_2\mathbf{R}_2 \\ . \\ . \\ \mathbf{Q}_p\mathbf{R}_p \end{pmatrix} = \mathtt{diag}(\mathbf{Q}_1,\dots,\mathbf{Q}_p) \begin{pmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ . \\ . \\ \mathbf{R}_p \end{pmatrix} , \qquad \mathbf{R}_{stack} = \begin{pmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ . \\ . \\ \mathbf{R}_p \end{pmatrix} = \mathbf{Q}_M\mathbf{R}_M$$

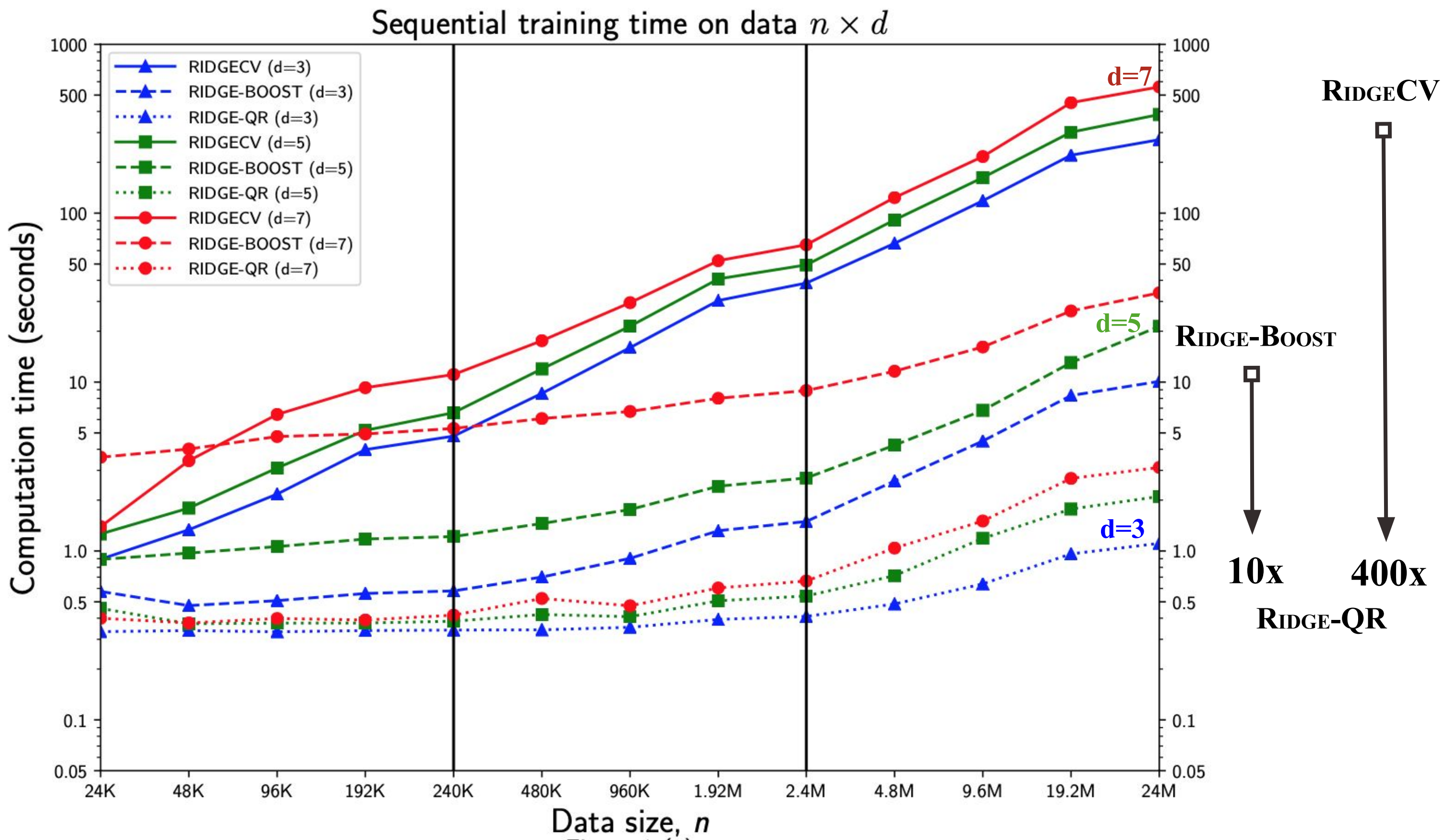
$$\mathbf{X} = \mathtt{diag}(\mathbf{Q}_1,\dots,\mathbf{Q}_p)\mathbf{R}_{stack} = \underbrace{\mathtt{diag}(\mathbf{Q}_1,\dots,\mathbf{Q}_p)\mathbf{Q}_M}_{\mathbf{Q}}\underbrace{\mathbf{R}_M}_{\mathbf{R}}$$



memory consumption per worker: $O(nd/p)$ | $O(d^2)$ | $O(pd^2)$ | $O(d^2)$

p: #workers

Results (1/3)



Sequential Training Time (RIDGE-QR vs others)

Results (2/3)

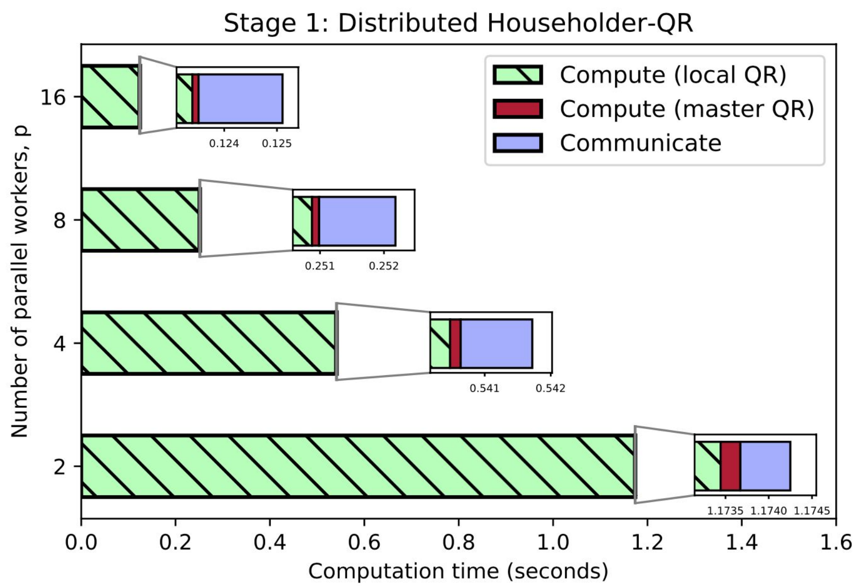


Figure 2 (a)

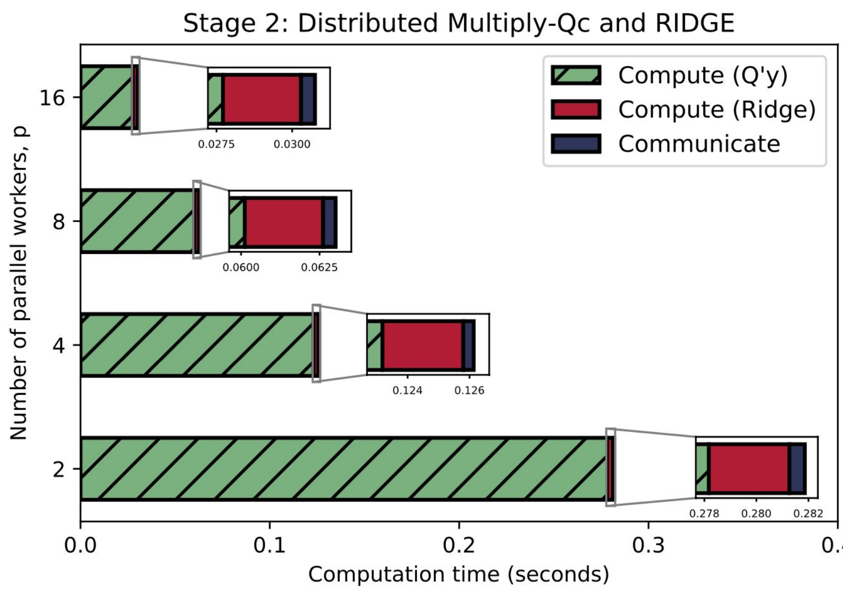
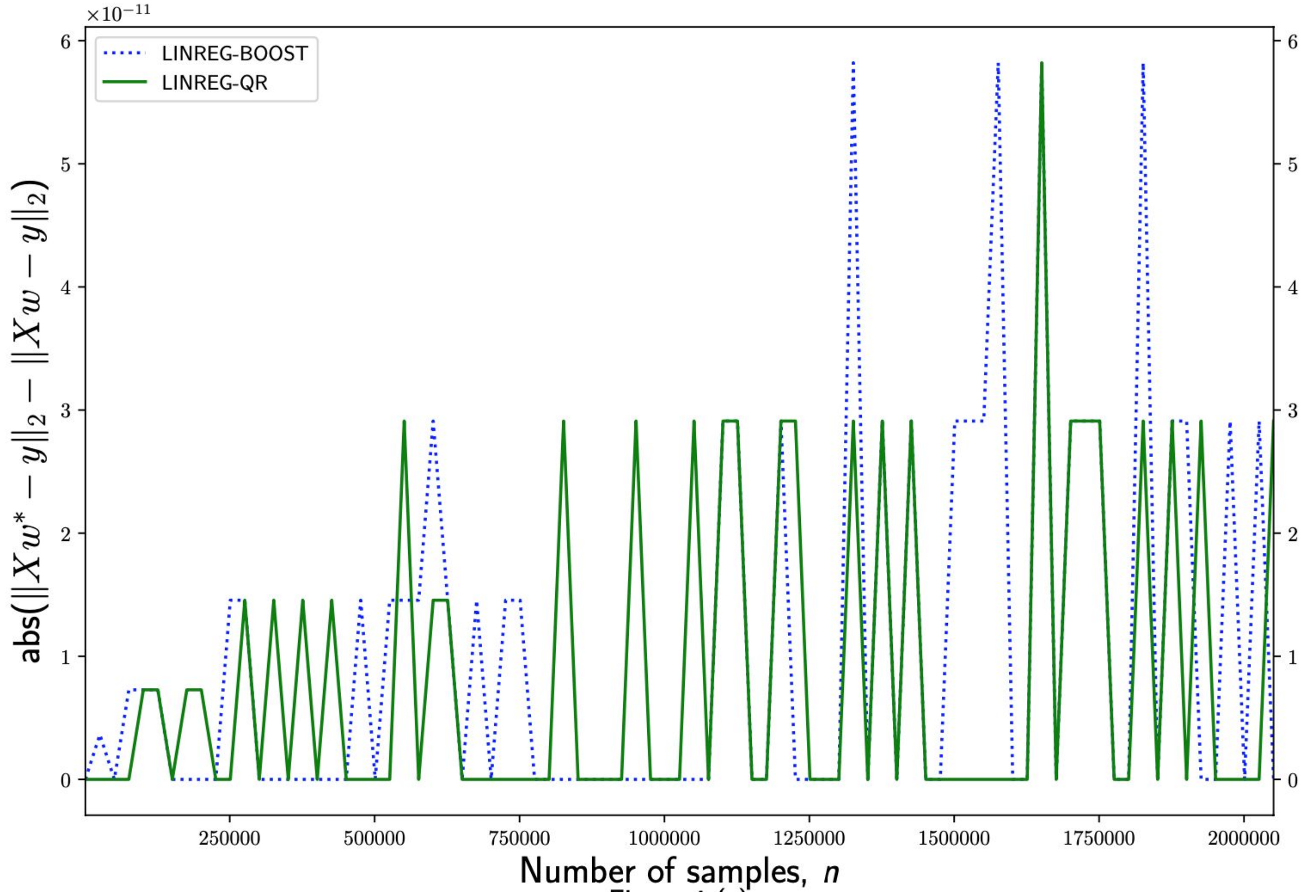


Figure 2 (b)

Execution Time breakdown of DISTRIBUTED RIDGE-QR (on 10M x 10)
with zoomed insets depicting communication time

Results (3/3)



Accuracy ($\times 10^{-11}$) comparison of LINREG-QR and LINREG-BOOST on Household Power Consumption dataset (~ 2M × 8), w* is solution from scikit-learn *LinearRegression*

Conclusions

Claim 1: QR decomposition is relatively time-consuming

FALSE

➔ Householder sketch is **more memory-efficient** and accelerates common LMS solvers in scikit-learn library **up to 100x-400x**, and outperforms the strong baseline LMS- BOOST by **10x-100x** with **similar numerical stability**.

Claim 2: QR decomposition is unsuitable for exact factorization for streaming data

FALSE

➔ The distributed implementation generates **accurate** distributed sketches and achieves **linear scalability** with **negligible communication** overhead for large sample size and dimension across multiple worker nodes.

Thank You !

Code

TEXAS A&M UNIVERSITY