A Deterministic Streaming Sketch for Ridge Regression

Benwei Shi
University of Utah

Jeff M. Phillips
University of Utah

Abstract

We provide a deterministic space-efficient algorithm for estimating ridge regression. For n data points with d features and a large enough regularization parameter, we provide a solution within ε L_2 error using only $O(d/\varepsilon)$ space. This is the first $o(d^2)$ space deterministic streaming algorithm with guaranteed solution error and risk bound for this classic problem. The algorithm sketches the covariance matrix by variants of Frequent Directions, which implies it can operate in insertion-only streams and a variety of distributed data settings. In comparisons to randomized sketching algorithms on synthetic and real-world datasets, our algorithm has less empirical error using less space and similar time.

1 INTRODUCTION

Linear regression is one of the canonical problems in machine learning. Given n pairs $(\mathbf{a}_i, b_i)$ with each $\mathbf{a}_i \in \mathbb{R}^d$ and $b \in \mathbb{R}$, we can accumulate them into a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ and vector $\mathbf{b} \in \mathbb{R}^n$. The goal is to find $\mathbf{x}_0 = \arg\min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$. It has a simple solution $\mathbf{x}_0 = \mathbf{A}^\dagger \mathbf{b}$ where $\mathbf{A}^\dagger$ is the pseudoinverse of $\mathbf{A}$. The most common robust variant, ridge regression (Hoerl and Kennard, 1970), uses a regularization parameter $\gamma > 0$ to add a squared ℓ_2 regularizer on $\mathbf{x}$. Its goal is

$$\mathbf{x}_\gamma = \arg\min_{\mathbf{x} \in \mathbb{R}^d} \left(\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 + \gamma \|\mathbf{x}\|^2 \right).$$

This also has simple solutions as

$$\mathbf{x}_\gamma = \begin{cases} (\mathbf{A}^\top \mathbf{A} + \gamma \mathbf{I})^{-1} \mathbf{A}^\top \mathbf{b}, & \text{when } n \geq d, \\ \mathbf{A}^\top (\mathbf{A}\mathbf{A}^\top + \gamma \mathbf{I})^{-1} \mathbf{b}, & \text{when } n \leq d, \end{cases}$$

where $\mathbf{I}$ is the identity matrix. The regularization and using $\gamma\mathbf{I}$ makes regression robust to noise (by reducing the variance), improves generalization, and avoids ill-conditioning.

However, this problem is difficult under very large data settings because the inverse operation and standard matrix multiplication will take $O(d^3 + nd^2)$ time, which is $O(nd^2)$ under our assumption $n > d$. And this can also be problematic if the size of $\mathbf{A}$, at $O(nd)$ space, exceeds memory. In a stream this can be computed in $O(d^2)$ space by accumulating $\mathbf{A}^\top \mathbf{A} = \sum_i \mathbf{a}_i^\top \mathbf{a}_i$ and $\mathbf{A}^\top \mathbf{b} = \sum_i \mathbf{a}_i^\top b_i$.

1.1 Previous Sketches

As a central task in data analysis, significant effort has gone into improving the running time of least squares (ridge) regression. Most improvements are in the form of sketching methods using projection or sampling. Sarlos (2006) initiated the formal study of using Random Projections (RP) for regression to reduce n dimensions to ℓ dimensions (still $\ell > d$) preserving the norm of the d dimension subspace vectors with high probability. Clarkson and Woodruff (2013) extended this technique to runtime depending on the number-of-non-zeroes, for sparse inputs, with CountSketch (CS). In non-streaming settings, the space can be reduced to depend on the rank $r = \mathrm{rank}(\mathbf{A})$ in the place of the full dimension. Lu et al. (2013) used a different random linear transform, called SRHT, and the dependence on the error was improved by Chen et al. (2015).

These random linear transform methods need a randomly selected subspace embedding with dimension ℓ, and the resulting sketches have size $O(\ell d)$. In the resulting analysis, the value ℓ should be greater than d or (if not streaming) r. If one strictly adheres to this theory, the large space bounds make the methods impractical when d is large and/or when requiring a high degree of accuracy (i.e., with small error parameter ε). One could of course still use the above methods to project to a small dimension with $\ell < d$ (as we do in our experiments), but no guarantees are known.

McCurdy (2018) proposed deterministic but not

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

streaming ridge leverage score sampling. Cohen et al. (2016, 2017) proposed streaming but not deterministic ridge leverage score sampling, relying on sketching techniques like Frequent Directions. In particular, their algorithms are strictly more complicated than the ones we will present, relying on additional randomized steps (ridge leverage score sampling) and analysis beyond the techniques we will employ. In particular, the computation of leverage scores depends on $(\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}$, which is also the key for the solution of ridge regression. These approaches can provide "risk" bounds (defined formally later), where the expected solution error is bounded under a Gaussian noise assumption. Recently, Wang et al. (2018) reanalyzed the quality of these previous linear ridge regression sketches from two related views: the optimization view (errors on objective function $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2+n\gamma\|\mathbf{x}\|^2$) and the statistics view (bias and variance of the solutions $\mathbf{x}$), but this work does not specifically improve the space or streaming analysis we focus on.

Although some of these sketches can be made streaming, if they use $o(d^2)$ space (so beating the simple $O(d^2)$ approach), they either do not provably approximate the solution coefficients, or are not streaming. And no existing streaming $o(d^2)$ space algorithm with any provable accuracy guarantees is deterministic.

1.2 Our Results

We make the observation, that if the goal is to approximate the solution to ridge regression, instead of ordinary least squares regression, and the regularization parameter is large enough, then a Frequent-Directions-based sketch (which only requires a single streaming pass) can preserve $(1 \pm \varepsilon)$-relative error on the solution parameters with only roughly $\ell = O(1/\varepsilon)$ rows. Thus it uses only $O(d\ell) = O(d/\varepsilon) = o(d^2)$ space. In contrast, streaming methods based on random linear transforms require $\ell = \Omega(1/\varepsilon^2)$ for similar guarantees. We formalize and prove this (see Theorems 4 and 5 for more nuanced statements), show evidence that this cannot be improved, and demonstrate empirically that indeed the FD-based sketch can significantly outperform random-projection-based sketches – especially in the space/error trade-off.

2 FREQUENT DIRECTIONS

Liberty (2013) introduced Frequent Directions (FD), then together with Ghashami et al. (2016b) improved the analysis. It considers a tall matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ (with $n \gg d$) row by row in a stream. It uses limited space $O(\ell d)$ to compute a short sketch matrix $\mathbf{B} \in \mathbb{R}^{\ell\times d}$, such that the covariance error is relatively small compared to the optimal rank k approximation, $\left\|\mathbf{A}^\top\mathbf{A} - \mathbf{B}^\top\mathbf{B}\right\|_2 \le \varepsilon \|\mathbf{A}-\mathbf{A}_k\|_F^2$. The algorithm maintains a sketch matrix $\mathbf{B} \in \mathbb{R}^{\ell\times d}$ representing the approximate right singular values of $\mathbf{A}$, scaled by the singular values. Specifically, it appends a batch of $O(\ell)$ new rows to $\mathbf{B}$, computes the SVD of $\mathbf{B}$, subtracts the squared ℓth singular value from all squared singular values (or marks down to 0), and then updates $\mathbf{B}$ as the reduced first $(\ell-1)$ singular values and right singular vectors. After each update, $\mathbf{B}$ has at most $\ell-1$ rows. After all rows of $\mathbf{A}$, for all $k < \ell$:

$$\left\|\mathbf{A}^\top\mathbf{A} - \mathbf{B}^\top\mathbf{B}\right\|_2 \le \frac{1}{\ell-k}\|\mathbf{A}-\mathbf{A}_k\|_F^2. \qquad (1)$$

The running time is $O(nd\ell)$ and required space is $O(\ell d)$. By setting $\ell = k+1/\varepsilon$, it achieves $\varepsilon\|\mathbf{A}-\mathbf{A}_k\|_F^2$ covariance error, in time $O(nd(k+1/\varepsilon))$ and in space $O((k+1/\varepsilon)d)$. Observe that setting $\ell = \mathrm{rank}(A)+1$ achieves 0 error in the form stated above.

Recently, Luo et al. (2019) proposed Robust Frequent Direction (RFD). They slightly extend FD by maintaining an extra value $\alpha \ge 0$, which is half of the sum of all squared ℓth singular values. Adding α back to the covariance matrix results in a more robust solution and less error. For all $0 \le k < \ell$:

$$\left\|\mathbf{A}^\top\mathbf{A} - \mathbf{B}^\top\mathbf{B} - \alpha\mathbf{I}\right\|_2 \le \frac{1}{2(\ell-k)}\|\mathbf{A}-\mathbf{A}_k\|_F^2. \quad (2)$$

It has same running time and running space with FD in terms of ℓ. To guarantee the same error, RFD needs almost a factor 2 fewer rows $\ell = 1/(2\varepsilon)+k$.

Huang (2018) proposed a more complicated variant to separate n from $1/\varepsilon$ in the running time. The idea is two level sketching: not only sketch $\mathbf{B} \in \mathbb{R}^{3k\times d}$, but also sketch the removed part into $\mathbf{Q} \in \mathbb{R}^{1/\varepsilon\times d}$ via sampling. Note that for a fixed k, $\mathbf{B}$ has a fixed number of rows, only $\mathbf{Q}$ increases the number of rows to reduce the error bound, and the computation of $\mathbf{Q}$ is faster and more coarse than that of $\mathbf{B}$. With high probability, for a fixed k, the sketch $\mathbf{B}^\top\mathbf{B}+\mathbf{Q}^\top\mathbf{Q}$ achieves the error in (1) in time $O(ndk) + \tilde{O}(\varepsilon^{-3}d)$ using space $O((k+\varepsilon^{-1})d)$. By setting $\ell = 3k+1/\varepsilon$, the running time is $O(nkd) + \tilde{O}((\ell-k)^3 d)$ and the space is $O(\ell d)$

The Frequent Directions sketch has other nice properties. It can be extended to have runtime depend only on the number of nonzeros for sparse inputs (Ghashami et al., 2016a; Huang, 2018). Moreover, it applies to distributed settings where data is captured from multiple locations or streams. Then these sketches can be "merged" together (Ghashami et al., 2016b; Agarwal et al., 2012) without accumulating any more error than the single stream setting,

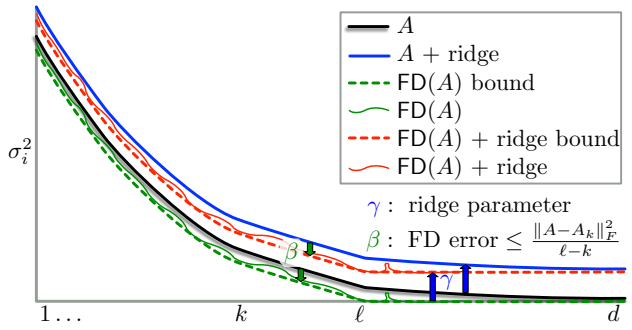

Figure 1: A figurative illustration of possible eigenvalues (σ_i^2) of a covariance matrices $\mathbf{A}^\top\mathbf{A}$ and variants when approximated by FD or adding a ridge term $\gamma\mathbf{I}$, along sorted eigenvectors.

and extend to other models (Shi et al., 2021). These properties apply directly to our new sketches.

2.1 FD and Ridge Regression

Despite FD being recognized as the matrix sketch with best space/error trade-off (often optimal (Ghashami et al., 2016b)), it has almost no provable connections improvements to high-dimensional regression tasks. The only previous approach we know of to connect FD to linear regression ((McCurdy, 2018) via (Cohen et al., 2017)), uses FD only to make the stream processing efficient, does not describe the actual algorithm, and then uses ridge leverage scores as an additional step to connect to ridge regression. The main challenge with connecting FD to linear regression is that FD approximates the high norm directions of $\mathbf{A}^\top\mathbf{A}$ (i.e., measured with direction/unit vector $\mathbf{x}$ as $\|\mathbf{A}^\top\mathbf{A}\mathbf{x}\|$), but drops the low norm directions. However, linear regression needs to recover $\mathbf{c} = \mathbf{A}^\top\mathbf{b}$ times the inverse of $\mathbf{A}^\top\mathbf{A}$. So if $\mathbf{c}$ is aligned with the low norm part of $\mathbf{A}^\top\mathbf{A}$, then FD provides a poor approximation. We observe however, that ridge regression with regularizer $\gamma\mathbf{I}$ ensures that all directions of $\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I}$ have norm at least γ, regardless of $\mathbf{A}$ or its sketch $\mathbf{B}$.

Figure 1 illustrates the effect on the eigenvalue distribution (as σ_i^2) for some $\mathbf{A}^\top\mathbf{A}$, and how it is affected by a ridge term and FD. The ridge term increases the values everywhere, and FD decreases the values everywhere. In principle, if these effects are balanced just right they should cancel out – at least for the high rank part of $\mathbf{A}^\top\mathbf{A}$. In particular, Robust Frequent Directions attempts to do this implicitly – it *automatically picks a good choice of regularizer* α as half of the amount of the shrinkage induced by FD.

3 ALGORITHMS AND ANALYSIS

We consider rows of $\mathbf{A} \in \mathbb{R}^{n\times d}$ and elements of $\mathbf{b} \in \mathbb{R}^n$ are given in pairs $(\mathbf{a}_i, b_i)$ in the stream, we want to approximate $\mathbf{x}_\gamma = (\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{b}$ for a given $\gamma > 0$ within space $O(\ell d)$, where $\ell < d$. Let $\mathbf{c} = \mathbf{A}^\top\mathbf{b}$, which can be exactly maintained using space $O(d)$. But $\mathbf{A}^\top\mathbf{A}$ needs space $\Omega(d^2)$, so we use Frequent Directions (FD) or Robust Frequent Directions (RFD) to approximate $\mathbf{A}$ by a sketch (which is an $\ell \times d$ matrix $\mathbf{C}$ and possibly also some auxiliary information). Then the optimal solution $\mathbf{x}_\gamma$ and its approximation of $\hat{\mathbf{x}}_\gamma$ are

$$\mathbf{x}_\gamma = (\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}\mathbf{c} \quad \text{and} \quad \hat{\mathbf{x}}_\gamma = (\text{sketch} + \gamma\mathbf{I})^{-1}\mathbf{c}.$$

Algorithm 1 General FD Ridge Regression (FDRR)

1: **Input:** $\ell, \mathbf{A}, \mathbf{b}, \gamma$
2: Initialize `xFD`, $\mathbf{c} \leftarrow 0^d$
3: **for** batches $(\mathbf{A}_\ell, \mathbf{b}_\ell) \in \mathbf{A}, \mathbf{b}$ **do**
4: sketch $\leftarrow$ `xFD`(sketch, $\mathbf{A}_\ell$)
5: $\mathbf{c} \leftarrow \mathbf{c} + \mathbf{A}_\ell^\top\mathbf{b}_\ell$
6: **end for**
7: $\hat{\mathbf{x}}_\gamma \leftarrow$ **Solution**(sketch, $\gamma, \mathbf{c}$)
8: **return** $\hat{\mathbf{x}}_\gamma$

Algorithm 1 shows the general algorithm framework. It processes a consecutive batch of ℓ rows of $\mathbf{A}$ (denoted $\mathbf{A}_\ell$) and ℓ elements of $\mathbf{b}$ (denoted $\mathbf{b}_\ell$) each step. `xFD` refers to a sketching step of some variant of Frequent Directions. Line 5 computes $\mathbf{A}^\top\mathbf{b}$ on the fly, it is not a part of FD. Line 7 computes the solution coefficients $\hat{\mathbf{x}}_\gamma$ using only the sketch of $\mathbf{A}$ and $\mathbf{c}$ at the end. This supplements FD with information to compute the ridge regression solution.

Coefficients error bound. The main part of our analysis is the upper bound of the *coefficients error*: $\varepsilon = \|\hat{\mathbf{x}}_\gamma - \mathbf{x}_\gamma\|/\|\mathbf{x}_\gamma\|$. Lemma 1 shows the key structural result, translating the sketch covariance error to the upper bound of ridge regression coefficients error.

Lemma 1. *Let* $\mathbf{C}^\top\mathbf{C}$ *be an approximation of* $\mathbf{A}^\top\mathbf{A} \in \mathbb{R}^{d\times d}$. *For any* $\mathbf{c} \in \mathbb{R}^d$, $\gamma \geq 0$, *consider an optimal solution* $\mathbf{x}_\gamma = (\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}\mathbf{c}$, *and an approximate solution* $\hat{\mathbf{x}}_\gamma = (\mathbf{C}^\top\mathbf{C} + \gamma\mathbf{I})^{-1}\mathbf{c}$. *Then*

$$\|\hat{\mathbf{x}}_\gamma - \mathbf{x}_\gamma\| \leq \frac{\|\mathbf{A}^\top\mathbf{A} - \mathbf{C}^\top\mathbf{C}\|_2}{\lambda_{min}(\mathbf{C}^\top\mathbf{C}) + \gamma}\|\mathbf{x}_\gamma\|.$$

Proof. To simplify the equations, let $\mathbf{M} = \mathbf{A}^\top\mathbf{A} + \lambda\mathbf{I}$, $\hat{\mathbf{M}} = \mathbf{C}^\top\mathbf{C} + \lambda\mathbf{I}$, then $\mathbf{M} - \hat{\mathbf{M}} = \mathbf{A}^\top\mathbf{A} - \mathbf{C}^\top\mathbf{C}$, and

so $\mathbf{x}_\gamma = \mathbf{M}^{-1}\mathbf{c}, \hat{\mathbf{x}}_\gamma = \hat{\mathbf{M}}^{-1}\mathbf{c}$.

$$\begin{aligned}\|\hat{\mathbf{x}}_\gamma - \mathbf{x}_\gamma\| &= \left\|\hat{\mathbf{M}}^{-1}\mathbf{c} - \mathbf{M}^{-1}\mathbf{c}\right\| = \left\|\left(\hat{\mathbf{M}}^{-1} - \mathbf{M}^{-1}\right)\mathbf{c}\right\| \\ &= \left\|\hat{\mathbf{M}}^{-1}\left(\mathbf{M} - \hat{\mathbf{M}}\right)\mathbf{M}^{-1}\mathbf{c}\right\| \\ &\le \left\|\hat{\mathbf{M}}^{-1}\right\|_2 \left\|\mathbf{M} - \hat{\mathbf{M}}\right\|_2 \left\|\mathbf{M}^{-1}\mathbf{c}\right\| \\ &= \frac{\|\mathbf{A}^\top\mathbf{A} - \mathbf{C}^\top\mathbf{C}\|_2}{\lambda_{\min}(\mathbf{C}^\top\mathbf{C}) + \gamma}\|\mathbf{x}_\gamma\|\end{aligned}$$

The third equality can be validated backwards by simple algebra. Here $\lambda_{\min}(\cdot)$ refer to the minimal eigenvalue of a matrix. □

Lemma 1 is tight when $\mathbf{A}^\top\mathbf{A} - \mathbf{C}^\top\mathbf{C} = \alpha\mathbf{I}$, and $\mathbf{C}^\top\mathbf{C} = \beta\mathbf{I}$ for any $\alpha, \beta \in \mathbb{R}$; see Lemma 2.

Lemma 2. *With the same settings as those in Lemma 1, if $\mathbf{A}^\top\mathbf{A} - \mathbf{C}^\top\mathbf{C} = \alpha\mathbf{I}$, and $\mathbf{C}^\top\mathbf{C} = \beta\mathbf{I}$ for any $\alpha, \beta \in \mathbb{R}$, then*

$$\|\hat{\mathbf{x}}_\gamma - \mathbf{x}_\gamma\| = \frac{\|\mathbf{A}^\top\mathbf{A} - \mathbf{C}^\top\mathbf{C}\|_2}{\lambda_{min}(\mathbf{C}^\top\mathbf{C}) + \gamma}\|\mathbf{x}_\gamma\|.$$

Proof. In the proof of Lemma 1, we have shown that $\|\hat{\mathbf{x}}_\gamma - \mathbf{x}_\gamma\| = \|\hat{\mathbf{M}}^{-1}\left(\mathbf{M} - \hat{\mathbf{M}}\right)\mathbf{M}^{-1}\mathbf{c}\|$, Using the definitions $\mathbf{M} = \mathbf{A}^\top\mathbf{A} + \lambda\mathbf{I}$, $\hat{\mathbf{M}} = \mathbf{C}^\top\mathbf{C} + \lambda\mathbf{I}$, and $\mathbf{x}_\gamma = (\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}\mathbf{c}$,

$$\begin{aligned}\|\hat{\mathbf{x}}_\gamma - \mathbf{x}_\gamma\| &= \|(\mathbf{C}^\top\mathbf{C} + \gamma\mathbf{I})^{-1}(\mathbf{A}^\top\mathbf{A} - \mathbf{C}^\top\mathbf{C})\mathbf{x}_\gamma\| \\ &= \|(\beta\mathbf{I} + \gamma\mathbf{I})^{-1}(\alpha\mathbf{I})\mathbf{x}_\gamma\| = \frac{\alpha}{\beta + \gamma}\|\mathbf{x}_\gamma\|.\end{aligned}$$

Similarly for the right hand side

$$\frac{\|\mathbf{A}^\top\mathbf{A} - \mathbf{C}^\top\mathbf{C}\|_2}{\lambda_{min}(\mathbf{C}^\top\mathbf{C}) + \gamma}\|\mathbf{x}_\gamma\| = \frac{\alpha}{\beta + \gamma}\|\mathbf{x}_\gamma\|. \qquad \square$$

Risk bound. We consider the fixed design setting commonly used in recent papers (Dhillon et al., 2013; Lu et al., 2013; Chen et al., 2015; McCurdy, 2018; Wang et al., 2018): we assume the data generation model is $\mathbf{b} = \mathbf{A}\mathbf{x} + s\boldsymbol{Z}$, where $\mathbf{A}, \mathbf{x}$ and s are fixed, $\boldsymbol{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the random error. The *risk* $\mathcal{R}(\hat{\mathbf{x}})$ of estimator $\hat{\mathbf{x}}$ of unknown coefficient $\mathbf{x}$ is the expected sum of squared error loss over the randomness of noise,

$$\mathcal{R}(\hat{\mathbf{x}}) = \mathbb{E}_{\boldsymbol{Z}}\left[\|\mathbf{A}\hat{\mathbf{x}} - \mathbf{A}\mathbf{x}\|^2\right] = \mathbb{E}_{\boldsymbol{Z}}\left[\|\mathbf{A}(\hat{\mathbf{x}} - \mathbf{x})\|^2\right].$$

We can further decompose the risk into *squared bias* and *variance*,

$$\begin{aligned}\mathcal{R}(\hat{\mathbf{x}}) &= \mathcal{B}^2(\hat{\mathbf{x}}) + \mathcal{V}(\hat{\mathbf{x}}), \\ \mathcal{B}^2(\hat{\mathbf{x}}) &= \|\mathbf{A}\left(\mathbb{E}_{\boldsymbol{Z}}[\hat{\mathbf{x}}] - \mathbf{x}\right)\|^2, \\ \mathcal{V}(\hat{\mathbf{x}}) &= \mathbb{E}_{\boldsymbol{Z}}\left[\|\mathbf{A}\left(\hat{\mathbf{x}} - \mathbb{E}_{\boldsymbol{Z}}[\hat{\mathbf{x}}]\right)\|^2\right].\end{aligned}$$

Lemma 3. *Given $\mathbf{A} \in \mathbb{R}^{n\times d}$, $\mathbf{x} \in \mathbb{R}^d$, $s > 0$, let $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ represent the standard Gaussian random variable, and $\mathbf{b} = \mathbf{A}\mathbf{x} + s\mathbf{Z}$, let $\mathbf{C}^\top\mathbf{C}$ be an deterministic approximation of $\mathbf{A}^\top\mathbf{A}$. Then the risk of optimal ridge regression solution $\mathbf{x}_\gamma = (\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{b}$ is the sum of*

$$\begin{aligned}\mathcal{B}^2(\mathbf{x}_\gamma) &= \gamma^2\left\|\mathbf{A}(\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}\mathbf{x}\right\|^2, \\ \mathcal{V}(\mathbf{x}_\gamma) &= s^2\|\mathbf{A}(\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}\mathbf{A}^\top\|_F^2.\end{aligned}$$

The risk of the approximate solution $\hat{\mathbf{x}}_\gamma = (\mathbf{C}^\top\mathbf{C} + \gamma\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{b}$ is the sum of

$$\begin{aligned}\mathcal{B}^2(\hat{\mathbf{x}}_\gamma) &= \left\|\mathbf{A}\left((\mathbf{C}^\top\mathbf{C} + \gamma\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{A} - \mathbf{I}\right)\mathbf{x}\right\|^2 \\ \mathcal{V}(\hat{\mathbf{x}}_\gamma) &= s^2\|\mathbf{A}(\mathbf{C}^\top\mathbf{C} + \gamma\mathbf{I})^{-1}\mathbf{A}^\top\|_F^2\end{aligned}$$

which are bounded as

$$\begin{aligned}\mathcal{B}^2(\hat{\mathbf{x}}_\gamma) &\le \left(1 + \frac{1}{\gamma^4}\|\mathbf{A}\|_2^4\|\mathbf{A}^\top\mathbf{A} - \mathbf{C}^\top\mathbf{C}\|^2\right)\mathcal{B}^2(\mathbf{x}_\gamma) \\ \mathcal{V}(\hat{\mathbf{x}}_\gamma) &\le (1 + \|\mathbf{A}\|_2^2/\gamma)^2\mathcal{V}(\mathbf{x}_\gamma).\end{aligned}$$

Proof. Within this proof, we sometimes use $\mathbf{K} = \mathbf{A}^\top\mathbf{A}$ and $\hat{\mathbf{K}} = \mathbf{C}^\top\mathbf{C}$ to shorten long equations.

Plugging $\mathbf{b} = \mathbf{A}\mathbf{x} + s\boldsymbol{Z}$ into $\mathbf{x}_\gamma = (\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{b}$ gives us

$$\mathbf{x}_\gamma = (\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{A}\mathbf{x} + (\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}\mathbf{A}^\top s\boldsymbol{Z}.$$

Since the standard Gaussian $\boldsymbol{Z}$ is the only random variable and we know that $\mathbb{E}_{\boldsymbol{Z}}[\mathbf{X}\boldsymbol{Z}] = 0$ for any $\mathbf{X} \in \mathbb{R}^{d\times n}$, thus

$$\mathbb{E}_{\boldsymbol{Z}}[\mathbf{x}_\gamma] = (\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{A}\mathbf{x}.$$

Similarly, we have

$$\hat{\mathbf{x}}_\gamma = (\mathbf{C}^\top\mathbf{C} + \gamma\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{A}\mathbf{x} + (\mathbf{C}^\top\mathbf{C} + \gamma\mathbf{I})^{-1}\mathbf{A}^\top s\boldsymbol{Z},$$

and

$$\mathbb{E}_{\boldsymbol{Z}}[\hat{\mathbf{x}}_\gamma] = (\mathbf{C}^\top\mathbf{C} + \gamma\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{A}\mathbf{x}.$$

By definition, the squared bias of $\mathbf{x}_\gamma$ is

$$\begin{aligned}\mathcal{B}^2(\mathbf{x}_\gamma) &= \|\mathbf{A}\left(\mathbb{E}_{\boldsymbol{Z}}[\mathbf{x}_\gamma] - \mathbf{x}\right)\|^2 \\ &= \left\|\mathbf{A}\left((\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{A}\mathbf{x} - \mathbf{x}\right)\right\|^2 \\ &= \left\|\mathbf{A}\left((\mathbf{K} + \gamma\mathbf{I})^{-1}\mathbf{K} - \mathbf{I}\right)\mathbf{x}\right\|^2 \\ &= \left\|\mathbf{A}\left((\mathbf{K} + \gamma\mathbf{I})^{-1}\mathbf{K} - (\mathbf{K} + \gamma\mathbf{I})^{-1}(\mathbf{K} + \gamma\mathbf{I})\right)\mathbf{x}\right\|^2 \\ &= \left\|\mathbf{A}\left((\mathbf{K} + \gamma\mathbf{I})^{-1}\left(\mathbf{K} - (\mathbf{K} + \gamma\mathbf{I})\right)\right)\mathbf{x}\right\|^2 \\ &= \left\|\mathbf{A}\left((\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}(-\gamma\mathbf{I})\right)\mathbf{x}\right\|^2 \\ &= \gamma^2\left\|\mathbf{A}(\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}\mathbf{x}\right\|^2.\end{aligned}$$

And the squared bias of $\hat{\mathbf{x}}_\gamma$ is

$$\begin{aligned}\mathcal{B}^2(\hat{\mathbf{x}}_\gamma) &= \|\mathbf{A}\left(\mathbb{E}_{\boldsymbol{Z}}[\hat{\mathbf{x}}_\gamma] - \mathbf{x}\right)\|^2 \\ &= \left\|\mathbf{A}\left((\mathbf{C}^\top\mathbf{C}+\gamma\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{A} - \mathbf{I}\right)\mathbf{x}\right\|^2.\end{aligned}$$

By playing with linear algebra, we can show that it is

$$\begin{aligned}&= \left\|\mathbf{A}\left((\hat{\mathbf{K}}+\gamma\mathbf{I})^{-1}\mathbf{K} - \mathbf{I}\right)\mathbf{x}\right\|^2 \\ &= \left\|\mathbf{A}\left(\left((\hat{\mathbf{K}}+\gamma\mathbf{I})^{-1} - (\mathbf{K}+\gamma\mathbf{I})^{-1} + (\mathbf{K}+\gamma\mathbf{I})^{-1}\right)\mathbf{K} - \mathbf{I}\right)\mathbf{x}\right\|^2 \\ &= \left\|\mathbf{A}\left(\left((\hat{\mathbf{K}}+\gamma\mathbf{I})^{-1}(\mathbf{K}-\hat{\mathbf{K}})(\mathbf{K}+\gamma\mathbf{I})^{-1} + (\mathbf{K}+\gamma\mathbf{I})^{-1}\right)\mathbf{K} - \mathbf{I}\right)\mathbf{x}\right\|^2 \\ &= \left\|\mathbf{A}\left((\hat{\mathbf{K}}+\gamma\mathbf{I})^{-1}(\mathbf{K}-\hat{\mathbf{K}})(\mathbf{K}+\gamma\mathbf{I})^{-1}\mathbf{K} + (\mathbf{K}+\gamma\mathbf{I})^{-1}\mathbf{K} - \mathbf{I}\right)\mathbf{x}\right\|^2 \\ &= \left\|\mathbf{A}\left((\hat{\mathbf{K}}+\gamma\mathbf{I})^{-1}(\mathbf{K}-\hat{\mathbf{K}})\mathbf{K}(\mathbf{K}+\gamma\mathbf{I})^{-1} - \gamma(\mathbf{K}+\gamma\mathbf{I})^{-1}\right)\mathbf{x}\right\|^2 \\ &= \left\|\left(\mathbf{A}(\hat{\mathbf{K}}+\gamma\mathbf{I})^{-1}(\mathbf{K}-\hat{\mathbf{K}})\mathbf{A}^\top\mathbf{A}(\mathbf{K}+\gamma\mathbf{I})^{-1} - \gamma\mathbf{A}(\mathbf{K}+\gamma\mathbf{I})^{-1}\right)\mathbf{x}\right\|^2 \\ &= \left\|\left(\tfrac{1}{\gamma}\mathbf{A}(\hat{\mathbf{K}}+\gamma\mathbf{I})^{-1}(\mathbf{K}-\hat{\mathbf{K}})\mathbf{A}^\top - \mathbf{I}\right)\gamma\mathbf{A}(\mathbf{K}+\gamma\mathbf{I})^{-1}\mathbf{x}\right\|^2 \\ &\le \left\|\tfrac{1}{\gamma}\mathbf{A}(\hat{\mathbf{K}}+\gamma\mathbf{I})^{-1}(\mathbf{K}-\hat{\mathbf{K}})\mathbf{A}^\top - \mathbf{I}\right\|^2 \gamma^2\left\|\mathbf{A}(\mathbf{K}+\gamma\mathbf{I})^{-1}\mathbf{x}\right\|^2 \\ &\le \left(\tfrac{1}{\gamma^2}\|\mathbf{A}\|_2^2\tfrac{1}{\gamma^2}\|\mathbf{K}-\hat{\mathbf{K}}\|^2\|\mathbf{A}\|_2^2 + 1\right)\gamma^2\left\|\mathbf{A}(\mathbf{K}+\gamma\mathbf{I})^{-1}\mathbf{x}\right\|^2 \\ &= \left(\frac{1}{\gamma^4}\|\mathbf{A}\|_2^4\|\mathbf{A}^\top\mathbf{A} - \mathbf{C}^\top\mathbf{C}\|^2 + 1\right)\mathcal{B}^2(\mathbf{x}_\gamma).\end{aligned}$$

The third equality follows $\hat{\mathbf{M}}^{-1} - \mathbf{M}^{-1} = \hat{\mathbf{M}}^{-1}(\mathbf{M} - \hat{\mathbf{M}})\mathbf{M}^{-1}$ for any invertable matrices $\mathbf{M}, \hat{\mathbf{M}}$ with the same dimensions, which has been used in the proof of Lemma 1. The fifth equality follows $(\mathbf{K}+\gamma\mathbf{I})^{-1}\mathbf{K} - \mathbf{I} = -\gamma(\mathbf{K}+\gamma\mathbf{I})^{-1}$, which has been shown in the derivation of $\mathcal{B}^2(\mathbf{x}_\gamma)$ above. The last inequality follows $\|(\hat{\mathbf{K}} + \gamma\mathbf{I})^{-1}\|_2^2 \le \frac{1}{\gamma^2}$ because $\hat{\mathbf{K}}$ is positive semi-definite.

For the variance part, by definition, the variance of $\mathbf{x}_\gamma$ is

$$\begin{aligned}\mathcal{V}(\mathbf{x}_\gamma) &= \mathbb{E}_{\boldsymbol{Z}}\left[\|\mathbf{A}\left(\mathbf{x}_\gamma - \mathbb{E}_{\boldsymbol{Z}}[\mathbf{x}_\gamma]\right)\|^2\right] \\ &= \mathbb{E}_{\boldsymbol{Z}}\left[\|\mathbf{A}\left((\mathbf{A}^\top\mathbf{A}+\gamma\mathbf{I})^{-1}\mathbf{A}^\top s\boldsymbol{Z}\right)\|^2\right] \\ &= s^2\|\mathbf{A}(\mathbf{A}^\top\mathbf{A}+\gamma\mathbf{I})^{-1}\mathbf{A}^\top\|_F^2.\end{aligned}$$

And the variance of $\hat{\mathbf{x}}_\gamma$ is

$$\begin{aligned}\mathcal{V}(\hat{\mathbf{x}}_\gamma) &= \mathbb{E}_{\boldsymbol{Z}}\left[\|\mathbf{A}\left(\hat{\mathbf{x}}_\gamma - \mathbb{E}_{\boldsymbol{Z}}[\hat{\mathbf{x}}_\gamma]\right)\|^2\right] \\ &= \mathbb{E}_{\boldsymbol{Z}}\left[\|\mathbf{A}\left((\mathbf{C}^\top\mathbf{C}+\gamma\mathbf{I})^{-1}\mathbf{A}^\top s\boldsymbol{Z}\right)\|^2\right] \\ &= s^2\|\mathbf{A}(\mathbf{C}^\top\mathbf{C}+\gamma\mathbf{I})^{-1}\mathbf{A}^\top\|_F^2 \\ &= s^2\|(\mathbf{A}^\dagger)^\dagger(\mathbf{C}^\top\mathbf{C}+\gamma\mathbf{I})^\dagger((\mathbf{A}^\top)^\dagger)^\dagger\|_F^2 \\ &= s^2\|((\mathbf{A}^\top)^\dagger\mathbf{C}^\top\mathbf{C}\mathbf{A}^\dagger + \gamma(\mathbf{A}^\top)^\dagger\mathbf{A}^\dagger)^\dagger\|_F^2 \\ &\le s^2\|(\gamma(\mathbf{A}\mathbf{A}^\top)^\dagger)^\dagger\|_F^2 = \frac{1}{\gamma^2}s^2\|\mathbf{A}^\top\mathbf{A}\|_F^2 \\ &\le \left(\frac{\|\mathbf{A}\|_2^2+\gamma}{\gamma}\right)^2\mathcal{V}(\mathbf{x}_\gamma) \\ &= (1 + \|\mathbf{A}\|_2^2/\gamma)^2\mathcal{V}(\mathbf{x}_\gamma).\end{aligned}$$

The fifth equality need the assumption that $\mathbf{A}$ has full column rank. The last inequality holds because

$$\begin{aligned}\mathcal{V}(\mathbf{x}_\gamma) &= s^2\sum_{i=1}^d\left(\frac{\sigma_i^2}{\sigma_i^2+\gamma}\right)^2 \ge s^2\sum_{i=1}^d\left(\frac{\sigma_i^2}{\sigma_1^2+\gamma}\right)^2 \\ &= \frac{s^2}{(\|\mathbf{A}\|_2^2+\gamma)^2}\sum_{i=1}^d\sigma_i^4 = \frac{s^2}{(\|\mathbf{A}\|_2^2+\gamma)^2}\|\mathbf{A}^\top\mathbf{A}\|_F^2.\end{aligned}$$

Here σ_i represent the ith singular value of $\mathbf{A}$. □

Note that the variance bound is independent of $\mathbf{C}^\top\mathbf{C}$; this is because it is positive definite and constructed deterministically. We also get some other variance bounds, Lemma 6 and 7 in the the Supplement Materials, which are related to the spectral bound, but can be much worse when $\|\mathbf{A}^\top\mathbf{A} - \mathbf{C}^\top\mathbf{C}\|_2^2 \neq 0$.

3.1 Using Frequent Directions

Now we consider Algorithm 2 (FDRR), using FD as `xFD` in Algorithm 1. Specifically, it uses the Fast Frequent Directions algorithm (Ghashami et al., 2016b). We explicitly store the first ℓ singular values $\boldsymbol{\Sigma}$ and singular vectors $\mathbf{V}^\top$, instead of $\mathbf{B}$, to be able to compute the the solution efficiently. Note that in the original FD algorithm, $\mathbf{B} = \boldsymbol{\Sigma}_\ell\mathbf{V}_\ell^\top$. Line 4 and 5 are what FD actually does in each step. It appends new rows $\mathbf{A}_\ell$ to the current sketch $\boldsymbol{\Sigma}_\ell\mathbf{V}_\ell^\top$, calls SVD to calculate the singular values $\boldsymbol{\Sigma}'$ and right singular vectors $\mathbf{V}'^\top$, then reduces the rank to ℓ.

Algorithm 2 Frequent Directions Ridge Regression (FDRR)

1: **Input:** $\ell, \mathbf{A}, \mathbf{b}, \gamma$
2: $\boldsymbol{\Sigma} \leftarrow 0^{\ell\times\ell}, \mathbf{V}^\top \leftarrow 0^{\ell\times d}, \mathbf{c} \leftarrow 0^d$
3: **for** batches $(\mathbf{A}_\ell, \mathbf{b}_\ell) \in \mathbf{A}, \mathbf{b}$ **do**
4: $\quad \_, \boldsymbol{\Sigma}', \mathbf{V}'^\top \leftarrow \text{SVD}([\mathbf{V}\boldsymbol{\Sigma}^\top; \mathbf{A}_\ell^\top]^\top)$
5: $\quad \boldsymbol{\Sigma} \leftarrow \sqrt{\boldsymbol{\Sigma}_\ell'^2 - \sigma_{\ell+1}^2\mathbf{I}_\ell}, \mathbf{V} \leftarrow \mathbf{V}_\ell'$
6: $\quad \mathbf{c} \leftarrow \mathbf{c} + \mathbf{A}_\ell^\top\mathbf{b}_\ell$
7: **end for**
8: $\mathbf{c}' \leftarrow \mathbf{V}^\top\mathbf{c}$
9: $\hat{\mathbf{x}}_\gamma \leftarrow \mathbf{V}(\boldsymbol{\Sigma}^2 + \gamma\mathbf{I}_\ell)^{-1}\mathbf{c}' + \gamma^{-1}(\mathbf{c} - \mathbf{V}\mathbf{c}')$
10: **return** $\hat{\mathbf{x}}_\gamma$

Line 8 and 9 are how we compute the solution $\hat{\mathbf{x}}_\gamma = (\mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^\top + \gamma\mathbf{I})^{-1}\mathbf{c}$. Explicitly inverting that matrix is not only expensive but also would use $O(d \times d)$ space, which exceeds the space limitation $O(\ell \times d)$. The good news is that $\mathbf{V}$ contains the eigenvectors of $(\mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^\top + \gamma\mathbf{I})^{-1}$, the corresponding ℓ eigenvalues $(\sigma_i^2 + \gamma)^{-1}$ for $i \in \{1, ..., \ell\}$, and the remaining eigenvalues are γ^{-1}. So we can separately compute $\hat{\mathbf{x}}_\gamma$ in the subspace spanned by $\mathbf{V}$ and its null space.

Theorem 4. *Given* $\mathbf{A} \in \mathbb{R}^{n\times d}$, $\mathbf{b} \in \mathbb{R}^n$, *let* $\mathbf{x}_\gamma = (\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{b}$ *and* $\hat{\mathbf{x}}_\gamma$ *be the output of Algorithm 2* $\textsc{FDrr}(\ell, \mathbf{A}, \mathbf{b}, \gamma)$. *If*

$$\ell \geq \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{\gamma\varepsilon} + k, \quad \text{or} \quad \gamma \geq \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{\varepsilon(\ell-k)},$$

then

$$\|\hat{\mathbf{x}}_\gamma - \mathbf{x}_\gamma\| \leq \varepsilon\|\mathbf{x}_\gamma\|.$$

It also holds that $\|\langle\hat{\mathbf{x}}_\gamma, \mathbf{a}'\rangle - \langle\mathbf{x}_\gamma, \mathbf{a}'\rangle\| \leq \varepsilon\|\mathbf{x}_\gamma\|\|\mathbf{a}'\|$ *for any* $\mathbf{a}' \in \mathbb{R}^d$, *and* $\|\mathbf{A}'\hat{\mathbf{x}}_\gamma - \mathbf{A}'\mathbf{x}_\gamma\| \leq \varepsilon\|\mathbf{x}_\gamma\|\|\mathbf{A}'\|_2$ *for any* $\mathbf{A}' \in \mathbb{R}^{m\times d}$. *The squared statistical bias* $\mathcal{B}^2(\hat{\mathbf{x}}_\gamma) \leq \left(1 + \frac{\varepsilon^2}{\gamma^2}\|\mathbf{A}\|_2^4\right)\mathcal{B}^2(\mathbf{x}_\gamma)$, *and the statistical variance* $\mathcal{V}(\hat{\mathbf{x}}_\gamma) \leq (1 + \|\mathbf{A}\|_2^2/\gamma)^2\mathcal{V}(\mathbf{x}_\gamma)$. *The running time is* $O(n\ell d)$ *and requires space* $O(\ell d)$.

Proof. Line 6 computes $\mathbf{c} = \mathbf{A}^\top\mathbf{b}$ in time $O(nd)$ using space $O(\ell d)$. Thus $\mathbf{x}_\gamma = (\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}\mathbf{c}$.

Line 8 and 9 compute the solution $\hat{\mathbf{x}}_\gamma = \mathbf{V}(\mathbf{\Sigma}^2 + \gamma\mathbf{I}_\ell)^{-1}\mathbf{V}^\top\mathbf{c} + \gamma^{-1}(\mathbf{c} - \mathbf{V}\mathbf{V}^\top\mathbf{c})$ in time $O(d\ell)$ using space $O(d\ell)$. Let $\mathbf{N} \in \mathbb{R}^{d\times(d-\ell)}$ be a set of orthonormal basis of the null space of $\mathbf{V}$. Then

$$\begin{aligned}
&\left(\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^\top + \gamma\mathbf{I}\right)^{-1} \\
&= \left(\begin{bmatrix}\mathbf{V} & \mathbf{N}\end{bmatrix}\begin{bmatrix}\mathbf{\Sigma}^2+\gamma\mathbf{I}_\ell & 0\\ 0 & \gamma\mathbf{I}_{d-\ell}\end{bmatrix}\begin{bmatrix}\mathbf{V} & \mathbf{N}\end{bmatrix}^\top\right)^{-1} \\
&= \begin{bmatrix}\mathbf{V} & \mathbf{N}\end{bmatrix}\begin{bmatrix}(\mathbf{\Sigma}^2+\gamma\mathbf{I}_\ell)^{-1} & 0\\ 0 & \gamma^{-1}\mathbf{I}_{d-\ell}\end{bmatrix}\begin{bmatrix}\mathbf{V} & \mathbf{N}\end{bmatrix}^\top \\
&= \mathbf{V}\left(\mathbf{\Sigma}^2+\gamma\mathbf{I}_\ell\right)^{-1}\mathbf{V}^\top + \mathbf{N}\left(\gamma^{-1}\mathbf{I}_{d-\ell}\right)\mathbf{N}^\top \\
&= \mathbf{V}\left(\mathbf{\Sigma}^2+\gamma\mathbf{I}_\ell\right)^{-1}\mathbf{V}^\top + \gamma^{-1}\mathbf{N}\mathbf{N}^\top \\
&= \mathbf{V}\left(\mathbf{\Sigma}^2+\gamma\mathbf{I}_\ell\right)^{-1}\mathbf{V}^\top + \gamma^{-1}\left(\mathbf{I} - \mathbf{V}\mathbf{V}^\top\right).
\end{aligned}$$

Thus $\hat{\mathbf{x}}_\gamma = (\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^\top + \gamma\mathbf{I})^{-1}\mathbf{c}$.

The rest of Algorithm 2 is equivalent to a normal FD algorithm with $\mathbf{B} = \mathbf{\Sigma}\mathbf{V}^\top$. Thus $\hat{\mathbf{x}}_\gamma = (\mathbf{B}^\top\mathbf{B} + \gamma\mathbf{I})^{-1}\mathbf{c}$, and satisfies (1). Together with Lemma 1 and $\lambda_{min}(\mathbf{B}^\top\mathbf{B}) \geq 0$, we have

$$\|\hat{\mathbf{x}}_\gamma - \mathbf{x}_\gamma\| \leq \frac{\|\mathbf{A}^\top\mathbf{A} - \mathbf{B}^\top\mathbf{B}\|_2}{\lambda_{min}(\mathbf{B}^\top\mathbf{B}) + \gamma}\|\mathbf{x}_\gamma\| \leq \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{\gamma(\ell-k)}\|\mathbf{x}_\gamma\|.$$

By setting $\frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{\gamma(\ell-k)} = \varepsilon$ and solving ℓ or γ, we get the guarantee for coefficients error. Plugging the FD result (1) into Lemma 3 gives us the risk bound. The running time and required space of a FD algorithm is $O(n\ell d)$ and $O(\ell d)$. Therefore the total running time is $O(nd) + O(\ell d) + O(n\ell d) = O(n\ell d)$, and the running space is $O(\ell d) + O(\ell d) + O(\ell d) = O(\ell d)$. □

Interpretation of bounds. Note that the only two approximations in the analysis of Theorem 4 arise from Lemma 1 and in the Frequent Directions bound. Both bounds are individually tight (see Lemma 2, and Theorem 4.1 by Ghashami et al. (2016b)), so while this is not a complete lower bound, it indicates this analysis approach cannot be asymptotically improved.

We can also write the space directly for this algorithm to achieve $\|\hat{\mathbf{x}} - \mathbf{x}_\gamma\| \leq \varepsilon\|\mathbf{x}_\gamma\|$ as $O(d(k + \frac{1}{\varepsilon}\frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{\gamma}))$. Note that this holds for all choices of $k < \ell$, so the space is actually $O(d \cdot \min_{0<k<\ell}(k + \frac{1}{\varepsilon}\frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{\gamma}))$. So when $\gamma = \Omega(\|\mathbf{A}-\mathbf{A}_k\|_F^2)$ (for an identified best choice of k) then this uses $O(d(k+\frac{1}{\varepsilon}))$ space, and if this holds for a constant k, then the space is $O(d/\varepsilon)$. This identifies the "regularizer larger than tail" case as when this algorithm is in theory appropriate. Empirically we will see below that it works well more generally.

3.2 Using Robust Frequent Directions

If we use RFD instead of FD, we store α in addition to $\mathbf{B} = \mathbf{\Sigma}\mathbf{V}^\top$; see Algorithm 3. Then the approximation of $\mathbf{A}^\top\mathbf{A}$ is $\mathbf{B}^\top\mathbf{B} + \alpha\mathbf{I} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^\top + \alpha\mathbf{I}$. We approximate $\mathbf{x}_\gamma$ by $\hat{\mathbf{x}}_\gamma = \left(\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^\top + (\gamma+\alpha)\mathbf{I}\right)^{-1}\mathbf{c}$. Line 6 in Algorithm 3 is added to maintain $\gamma + \alpha$. The remainder of the algorithm is the same as Algorithm 2. The theoretical results slightly improve those for FD. Theorem 5 and its proof is established by replacing FD result with RFD result (2) in Theorem 4.

Algorithm 3 Robust Frequent Directions Ridge Regression (RFDrr)

1: **Input:** $\ell, \mathbf{A} \in ,\mathbf{b}, \gamma$
2: $\mathbf{\Sigma} \leftarrow 0^{\ell\times\ell}, \mathbf{V}^\top \leftarrow 0^{\ell\times d}, \mathbf{c} \leftarrow 0^d$
3: **for** $\mathbf{A}_\ell, \mathbf{b}_\ell \in \mathbf{A}, \mathbf{b}$ **do**
4: $\quad$ $\_, \mathbf{\Sigma}', \mathbf{V}'^\top \leftarrow \textsc{svd}([\mathbf{V}\mathbf{\Sigma}^\top; \mathbf{A}_\ell^\top]^\top)$
5: $\quad$ $\mathbf{\Sigma} \leftarrow \sqrt{\mathbf{\Sigma}_\ell'^2 - \sigma_{\ell+1}^2\mathbf{I}_\ell}, \mathbf{V} \leftarrow \mathbf{V}'_\ell$
6: $\quad$ $\gamma \leftarrow \gamma + \sigma_{\ell+1}^2/2$
7: $\quad$ $\mathbf{c} \leftarrow \mathbf{c} + \mathbf{A}_\ell^\top\mathbf{b}_\ell$
8: **end for**
9: $\mathbf{c}' \leftarrow \mathbf{V}^\top\mathbf{c}$
10: $\hat{\mathbf{x}}_\gamma \leftarrow \mathbf{V}(\mathbf{\Sigma}^2 + \gamma\mathbf{I}_\ell)^{-1}\mathbf{c}' + \gamma^{-1}(\mathbf{c} - \mathbf{V}\mathbf{c}')$
11: **return** $\hat{\mathbf{x}}_\gamma$

Theorem 5. *Given* $\mathbf{A} \in \mathbb{R}^{n\times d}$, $\mathbf{b} \in \mathbb{R}^n$, *let* $\mathbf{x}_\gamma = (\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{b}$ *and* $\hat{\mathbf{x}}_\gamma$ *be output of Algorithm 3 with input* $(\ell, \mathbf{A}, \mathbf{b}, \gamma)$. *If*

$$\ell \geq \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{2\gamma\varepsilon} + k, \quad \text{or} \quad \gamma \geq \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{2\varepsilon(\ell-k)}$$

then

$$\|\hat{\mathbf{x}}_\gamma - \mathbf{x}_\gamma\| \leq \varepsilon\|\mathbf{x}_\gamma\|$$

It also holds that $\|\langle\hat{\mathbf{x}}_\gamma, \mathbf{a}'\rangle - \langle\mathbf{x}_\gamma, \mathbf{a}'\rangle\| \leq \varepsilon\|\mathbf{x}_\gamma\|\|\mathbf{a}'\|$ *for any* $\mathbf{a}' \in \mathbb{R}^d$, *and* $\|\mathbf{A}'\hat{\mathbf{x}}_\gamma - \mathbf{A}'\mathbf{x}_\gamma\| \leq \varepsilon\|\mathbf{x}_\gamma\|\|\mathbf{A}'\|_2$ *for any* $\mathbf{A}' \in \mathbb{R}^{m\times d}$. *The squared statistical bias*

$\mathcal{B}^2(\hat{\mathbf{x}}_\gamma) \leq \left(1 + \frac{4\varepsilon^2}{\gamma^2}\|\mathbf{A}\|_2^4\right)\mathcal{B}^2(\mathbf{x}_\gamma)$, *and the statistical variance* $\mathcal{V}(\hat{\mathbf{x}}_\gamma) \leq (1 + \|\mathbf{A}\|_2^2/\gamma)\mathcal{V}(\mathbf{x}_\gamma)$. *The running time is* $O(n\ell d)$ *and requires space* $O(\ell d)$.

4 EXPERIMENTS

We compare new algorithms FDRR and RFDRR with other FD-based algorithms and randomized algorithms on synthetic and real-world datasets. We focus only on streaming algorithms.

Competing algorithms include:

- ISVDRR: Truncated incremental SVD (Brand, 2002; Hall et al., 1998), also known as Sequential Karhunen–Loeve (Levey and Lindenbaum, 2000), for sketching, has the same framework as Algorithm 2 but replaces Line 5 $\mathbf{\Sigma} \leftarrow \sqrt{\mathbf{\Sigma}_\ell'^2 - \sigma_{\ell+1}^2\mathbf{I}_\ell}, \mathbf{V} \leftarrow \mathbf{V}_\ell'$ with $\mathbf{\Sigma} \leftarrow \mathbf{\Sigma}_\ell', \mathbf{V} \leftarrow \mathbf{V}_\ell'$. That is, it simply maintains the best rank-ℓ approximation after each batch.
- 2LFDRR: This uses a two-level FD variant proposed by Huang (2018) for sketching, and described in more detail in Section 2.
- RPRR: This uses generic (scaled) {-1,+1} random projections (Sarlos, 2006). For each batch of data, construct a random matrix $\mathbf{S} \in \{-\sqrt{\ell}, \sqrt{\ell}\}^{\ell\times\ell}$, set $\mathbf{C} = \mathbf{C} + \mathbf{SA}$ and $\mathbf{c} = \mathbf{c} + \mathbf{Sb}$. Output $\hat{\mathbf{x}}_\gamma = (\mathbf{C}^\top\mathbf{C} + \gamma\mathbf{I})^{-1}\mathbf{C}^\top\mathbf{c}$ at the end.
- CSRR: This is the sparse version of RPRR using the CountSketch (Clarkson and Woodruff, 2013). The random matrix $\mathbf{S}$ are all zeros except for one -1 or 1 in each column with a random location.
- RR: This is the naive streaming ridge regression which computes $\mathbf{A}^\top\mathbf{A}$ and $\mathbf{A}^\top\mathbf{b}$ cumulatively (a batch size of 1). In each step it computes $\mathbf{A}^\top\mathbf{A} \leftarrow \mathbf{A}^\top\mathbf{A} + \mathbf{a}_i^\top\mathbf{a}_i$ where $\mathbf{a}_i^\top\mathbf{a}_i$ is an outer product of row vectors, and $\mathbf{c} \leftarrow \mathbf{c} + \mathbf{a}_i^\top b_i$. Then it outputs $\mathbf{x}_\gamma = (\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I})^{-1}\mathbf{c}$ at the end. This algorithm uses d^2 space and has no error in $\mathbf{A}^\top\mathbf{A}$ or $\mathbf{c}$. This algorithm's found ridge coefficients $\mathbf{x}_\gamma$ are used to compute the coefficients error of all sketching algorithms.

Datasets. We use three main datasets that all have dimension $d = 2^{11}$, training data size $n = 2^{13}$, and test data size $n_t = 2^{11}$.

Synthetic datasets. Two synthetic data-sets are low rank (LR) and high rank (HR), determined by an effective rank parameter R; set $R = \lfloor 0.1d \rfloor$ and $R = \lfloor 0.5d \rfloor$ respectively, which is 10 and 50 percent of d. This R is then used as the number of non-zero coefficients $\mathbf{x}$ and the number of major standard deviations of a multivariate normal distribution for generating input points $\mathbf{A}$. Each row vector of $\mathbf{A} \in \mathbb{R}^{n\times d}$ are generated by normal distribution with standard deviations $s_i = \exp(-\frac{i^2}{R^2})$ for $i = 0, 1, ..., d-1$, so the maximal standard deviation is $s_0 = 1$. Figure 2 shows the singular value distributions datasets, normalized by their first singular values, and indices normalized by d. The linear model coefficients $\mathbf{x} \in \mathbb{R}^d$ have first R entries non-zero, they are generated by another standard normal distribution, then normalized to a unit vector so the gradient of the linear model is 1. A Gaussian noise $\boldsymbol{Z} \sim \mathcal{N}(\mathbf{0}, 4\mathbf{I})$ is added to the outputs, i.e. $\mathbf{b} = \mathbf{Ax} + \boldsymbol{Z}$. Finally, we rotate $\mathbf{A}$ by a discrete cosine transform.

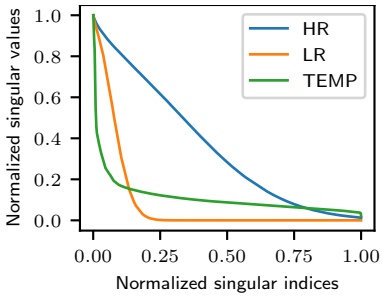

Figure 2: Datasets singular values

TEMP: Temperature sequence. This is derived from the temperature sequence recorded hourly from 1997 to 2019 at an international airport. To model an AR process, we compute the difference sequence between hourly temperatures, and then shingle this data, so $\mathbf{a}_i$ is d consecutive differences starting at the ith difference, and b_i is the next (the $(i+d)$th) difference between temperatures. Then the TEMP dataset matrix $\mathbf{A}$ is a set of n randomly chosen (without replacement) such shingles.

Choice of γ. We first run RR on training datasets with different γs, then choose the ones which best minimize $\|\mathbf{A}_{\text{test}}x_\gamma^* - \mathbf{b}_{\text{test}}\|$ using a held out test dataset $(\mathbf{A}_{\text{test}}, \mathbf{b}_{\text{test}})$. The best γs for low rank LR and high rank HR datasets are 4096 and 32768 respectively, the best γ for TEMP dataset is 32768. These γ values are fixed for the further experiments. Since the γ value is only used to compute the solution $\mathbf{x}_\gamma$ or $\hat{\mathbf{x}}$ (storing α separate from γ in RFDRR), so this choice could be made when calculating the solution using a stored test set after sketching. To avoid this extra level of confounding error into the evaluation process, we simply use this pre-computed γ value.

4.1 Evaluation

We run these 6 algorithms with different choices of ℓ on these three datasets. They are implemented in python using numpy, and are relatively straightforward. For completeness, we will release de-anonymized code and data for reproducibility after double-blind peer review. We first train them on the training sets, query their

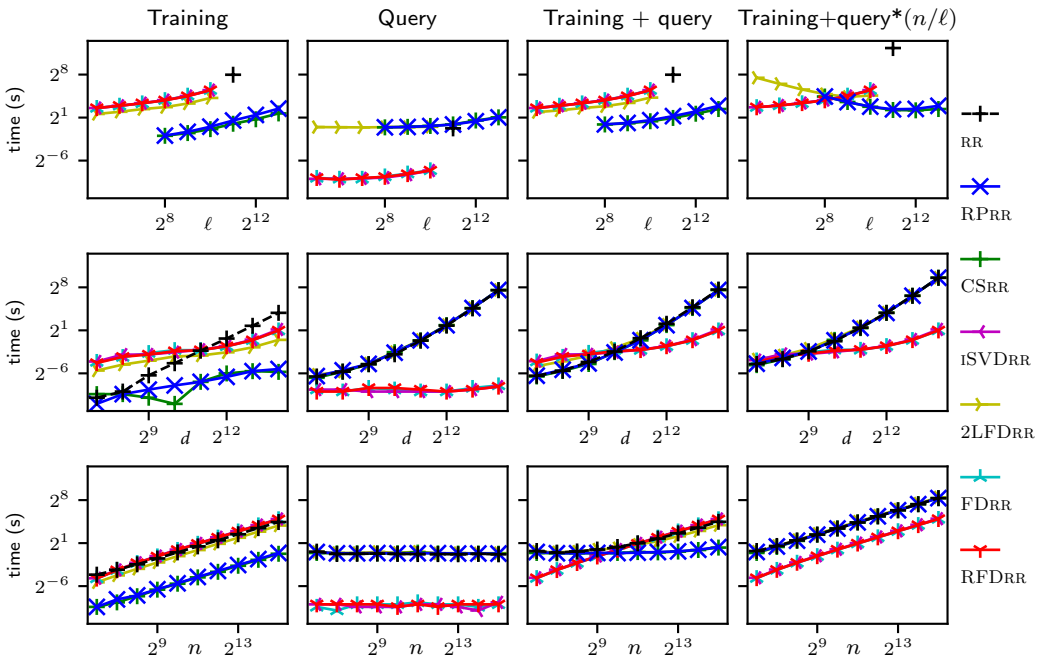

Figure 3: Running time (seconds) as a function of: sketch size parameter ℓ (Row 1), data dimension d (Row 2), and training set size n (Row 3).

coefficients, then compute the coefficients errors with RR and prediction errors with outputs. We repeat all these experiments 10 times and show the mean results.

Running time. In Figure 3, Row 1 we show the running time (on HR) by training time, solution query (computation of the coefficients) time, their sum, and training time + query time$*n/\ell$ simulating making a query every batch. The other datasets are the same size, and have the same runtimes. FD based algorithms are slower then randomized algorithms during training, but much faster during query solutions since the sketch sizes are smaller and more processed. They maintain the SVD results of the sketch so the matrix inversion is mostly precomputed. Note that this precomputation is not available in the two-level 2LFDRR either, hence this also suffers from higher query time.

When we add the training time and (n/ℓ) queries, then ISVDRR, RFDRR, and FDRR are the fastest for ℓ below about 300 (past 2^8). Note that in this plot the number of batches and hence queries decreases as ℓ increases, and as a result for small ℓ the algorithms with cost dominated by queries (CSRR, RPRR, and 2LFDRR) have their runtime initially decrease. All algorithms are generally faster than RR – the exception is the random projection algorithms (CSRR and RPRR) which are a bit slower for query time, and these become worse as ℓ becomes greater than d.

In Figure 3, Row 2 and 3 we show the runtime of the algorithms as both n and d increase. We fix $\ell = 2^6$. When we vary d we fix $n = 2^8$, and when we vary n we fix $d = 2^{11}$. As expected, the runtimes all scale linearly as n grows, or the sum of two linear times for (training+query) time. As d grows, FD-based algorithms (not including 2LFDRR) overcome RP-based algorithms (as well as RR and 2LFDRR) even with one query. The query time for the latter increase too fast, cubic on d, but is linear for FD-based algorithms.

Accuracy. Let $\mathbf{x}_\gamma$ be the coefficients solutions of RR and $\hat{\mathbf{x}}_\gamma$ be its approximation, let $\hat{\mathbf{b}}$ be the predicted values by RR, $\mathbf{A}\mathbf{x}_\gamma$, or its approximation, $\mathbf{A}\hat{\mathbf{x}}_\gamma$; for each algorithm we compute the coefficients error (coef. error $= \|\hat{\mathbf{x}}_\gamma - \mathbf{x}_\gamma\|/\|\mathbf{x}_\gamma\|$) and the prediction error (pred. error $= \|\hat{\mathbf{b}} - \mathbf{b}\|^2/n$). Figure 4 shows these errors versus space in terms of ℓ, and (training + $\frac{n}{\ell}$query) time in seconds. For the high rank data (top row), all FD-based algorithms (FDRR, RFDRR, 2LFDRR, as well as ISVDRR) have far less error than the random projection algorithms (RPRR and CSRR). For very small ℓ size RFDRR does slightly worse than the other FD variants, likely because it adds too much bias because the "tail" is too large with small ℓ.

For the low rank data and real-world TEMP data the errors are more spread out, yet the FD-based algorithms still do significantly better as a function of space (ℓ). Among these RFDRR (almost) always has the least error (for small ℓ) or matches the best error

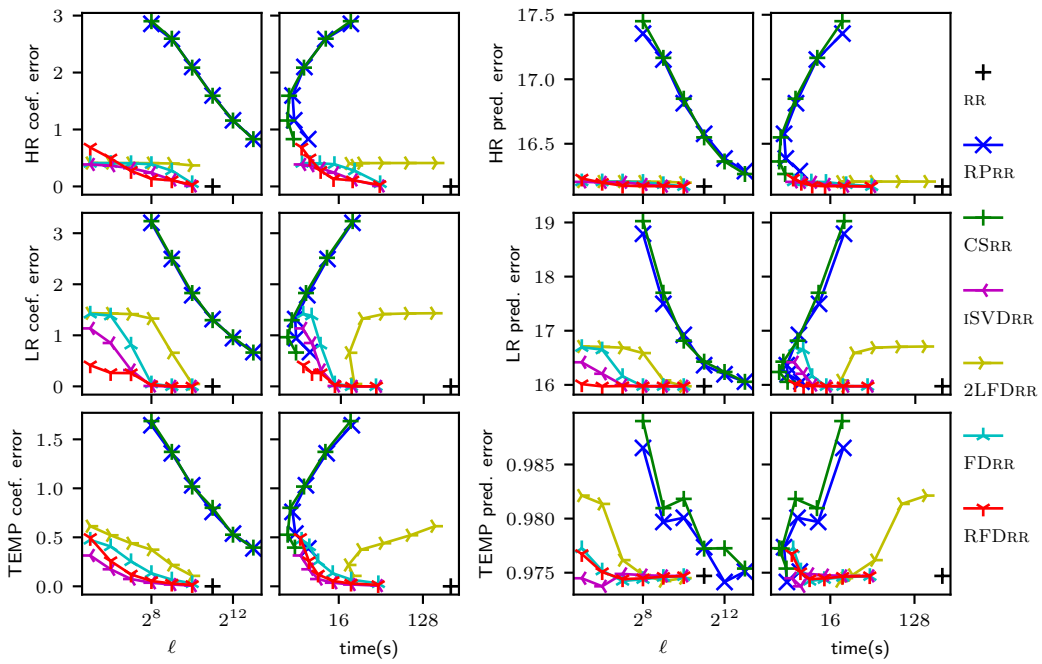

Figure 4: Errors vs space (measured by rows ℓ) and time (measured by seconds). The time shown is the training time + the query time$*\frac{n}{\ell}$ to simulate a query every batch. The left double column shows coefficient error, and the right double column shows prediction error. Note that the runtime for CSRR and RPRR form a 'C' shape since these are query-dominated, and the runtime initially decreases as the number of queries (number of "batches") decreases, as ℓ increases, like in Figure 3, Row 1.

(for larger ℓ). The only one that sometimes improves upon RFDRR, and is otherwise the next best is the huersistic iSVDRR which has no guarantees, and likely will fail for adversarial data (Desai et al., 2016). In terms of the time, the random projection algorithms can be a bit faster (say 4 seconds instead of $5-10$ seconds), but then achieve more coefficient error. In particular, RFDRR always can achieve the least coefficient error, and usually the least coefficient error for any given allotment of time. For prediction error as a function of time (the rightmost column of Figure 4), the results are more muddled. Many algorithms can achieve the minimum error (nearly matching RR) in the nearly best runtime (about $5-7$ seconds). The FD-based algorithms are roughly at this optimal points for all ℓ parameters tried above $\ell = 2^5$, and hence consistently achieves these results in small space and time.

5 CONCLUSION & DISCUSSION

We provide the first streaming sketch algorithms that can apply the optimally space efficient Frequent Directions sketch towards regression, focusing on ridge regression. This results in the first streaming deterministic sketch using $o(d^2)$ space in $\mathbb{R}^d$. We demonstrate that our bounds will be difficult to be improved, and likely cannot be. We also prove new risk bounds, comparable to previous results, but notably have a variance bound independent of the specific sketch matrix chosen. Similar to prior observations (McCurdy, 2018; Cohen et al., 2016), we show the ridge term makes regression easier to sketch. Moreover, our experiments demonstrate that while these FD-based algorithms have larger training time than random projection ones, they have less empirical error, their space usage is smaller, and query time is often far more efficient. Our proposed sketches clearly have the best space/error trade-off.

Discussion relating to PCR. Principal Component Regression (PCR) is a related approach; it identifies the top k principal components $\mathbf{V}_k$ of $\mathbf{A}$ and performs regression using, $[\pi_{\mathbf{V}_k}(\mathbf{A}), \mathbf{b}]$, the projection onto the span of $\mathbf{V}_k$. For this to be effective, these components must include the directions meaningfully correlated with $\mathbf{A}^\top\mathbf{b}$. However, when the top $k' > k$ singular vectors of $\mathbf{A}$ are all similar, which of the corresponding top k' singular vectors are in the top k is not stable. If a meaningful direction among the top-k is not retained in a top-k sketch $\mathbf{B}$, then while the norms of $\mathbf{A}$ are preserved using a sketch $\mathbf{B}$, the regression result may be quite different. Hence, PCR is not stable in the same way as RR, and precludes approximation guarantees in the strong form similar to ours.

Acknowledgements. Jeff M. Phillips thanks his support from NSF IIS-1816149, CCF-1350888, CNS-1514520, CNS-1564287, and CFS-1953350.

References

Pankaj K Agarwal, Graham Cormode, Zengfeng Huang, Jeff M. Phillips, Zhewei Wei, and Ke Yi. Mergeable summaries. In *Proceedings of the 31st symposium on Principles of Database Systems - PODS '12*, pages 23–34. ACM, 2012. ISBN 978-1-4503-1248-6. doi: 10.1145/2213556.2213562.

Matthew Brand. Incremental singular value decomposition of uncertain data with missing values. In *Computer Vision — ECCV 2002*, 2002.

Shouyuan Chen, Yang Liu, Michael R. Lyu, Irwin King, and Shengyu Zhang. Fast relative-error approximation algorithm for ridge regression. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, 2015.

Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, 2013.

Michael B. Cohen, Cameron Musco, and Jakub Pachocki. Online row sampling. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016)*, volume 60 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 2016.

Michael B. Cohen, Cameron Musco, and Christopher Musco. Input sparsity time low-rank approximation via ridge leverage score sampling. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '17, 2017.

A. Desai, M. Ghashami, and J. M. Phillips. Improved practical matrix sketching with guarantees. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1678–1690, 2016.

Paramveer S Dhillon, Dean P Foster, Sham M Kakade, and Lyle H Ungar. A Risk Comparison of Ordinary Least Squares vs Ridge Regression. *The Journal of Machine Learning Research*, 14:1505–1511, 2013.

Mina Ghashami, Edo Liberty, and Jeff M. Phillips. Efficient Frequent Directions Algorithm for Sparse Matrices. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, 2016a.

Mina Ghashami, Edo Liberty, Jeff M. Phillips, and David P. Woodruff. Frequent Directions: Simple and Deterministic Matrix Sketching. *SIAM Journal on Computing*, 45(5):1762–1792, 2016b.

Peter M. Hall, David Marshall, and Ralph R. Martin. Incremental eigenanalysis for classification. In *in British Machine Vision Conference*, 1998.

Arthur E. Hoerl and Robert W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55–67, feb 1970. ISSN 0040-1706. doi: 10.1080/00401706.1970.10488634. URL `http://www.tandfonline.com/doi/abs/10.1080/00401706.1970.10488634`.

Zengfeng Huang. Near optimal frequent directions for sketching dense and sparse matrices. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 2018.

A. Levey and M. Lindenbaum. Sequential karhunen-loeve basis extraction and its application to images. *IEEE Transactions on Image Processing*, 9(8):1371–1374, 2000.

Edo Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, 2013.

Yichao Lu, Paramveer Dhillon, Dean P Foster, and Lyle Ungar. Faster ridge regression via the subsampled randomized hadamard transform. In *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., 2013.

Luo Luo, Cheng Chen, Zhihua Zhang, Wu-Jun Li, and Tong Zhang. Robust Frequent Directions with Application in Online Learning. *Journal of Machine Learning Research*, 20(45):1–41, 2019.

Shannon R. McCurdy. Ridge regression and provable deterministic ridge leverage score sampling. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS'18, 2018.

T. Sarlos. Improved approximation algorithms for large matrices via random projections. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, 2006.

Benwei Shi, Zhuoyue Zhao, Yanqing Peng, Feifei Li, and Jeff M. Phillips. At-the-time and back-in-time persistent sketches. In *ACM Symposium on Management of Data (SIGMOD)*, 2021.

Shusen Wang, Alex Gittens, and Michael W. Mahoney. Sketched ridge regression: Optimization perspective, statistical perspective, and model averaging. *Journal of Machine Learning Research*, 18(218):1–50, 2018.