Fast Relative-Error Approximation Algorithm for Ridge Regression

Shouyuan Chen[1*] **Yang Liu**[21*] **Michael R. Lyu**[31] **Irwin King**[31] **Shengyu Zhang**[21]
3: Shenzhen Key Laboratory of Rich Media Big Data Analytics and Applications,
Shenzhen Research Institute, The Chinese University of Hong Kong, Shenzhen, China
2: The Institute of Theoretical Computer Science and Communications, The Chinese University of Hong Kong
1: Department of Computer Science and Engineering, The Chinese University of Hong Kong
chenshouyuan@gmail.com {yliu,lyu,king,syzhang}@cse.cuhk.edu.hk

Abstract

Ridge regression is one of the most popular and effective regularized regression methods, and one case of particular interest is that the number of features p is much larger than the number of samples n, i.e. $p \gg n$. In this case, the standard optimization algorithm for ridge regression computes the optimal solution $\mathbf{x}^*$ in $O(n^2p + n^3)$ time. In this paper, we propose a fast *relative-error* approximation algorithm for ridge regression. More specifically, our algorithm outputs a solution $\tilde{\mathbf{x}}$ satisfying $\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_2 \le \epsilon \|\mathbf{x}^*\|_2$ with high probability and runs in $\tilde{O}(\mathrm{nnz}(\mathbf{A}) + n^3/\epsilon^2)$ time, where $\mathrm{nnz}(\mathbf{A})$ is the number of non-zero entries of matrix $\mathbf{A}$.

To the best of our knowledge, this is the first algorithm for ridge regression that runs in $o(n^2p)$ time with provable relative-error approximation bound on the output vector. In addition, we analyze the risk inflation bound of our algorithm and apply our techniques to two generalizations of ridge regression, including multiple response ridge regression and a non-linear ridge regression problem. Finally, we show empirical results on both synthetic and real datasets.

1 INTRODUCTION

Ridge regression is one of the most popular and effective regularized regression methods, and one case of particular interest is that the number of features p is much larger than the number of samples n, i.e. $p \gg n$. The definition of ridge regression problem is as follows. Given an $n \times p$ sample-by-feature design matrix $\mathbf{A}$ together with an n-dimensional target vector $\mathbf{b}$, and a parameter $\lambda > 0$, the goal of ridge regression is to find a p-dimensional vector $\mathbf{x}^*$ such that

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathbb{R}^p} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_2^2 . \qquad (1)$$

Saunders et al. [29] showed that that the unique minimizer of Eq. (1) can be computed as follows

$$\mathbf{x}^* = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T + \lambda \mathbf{I}_n)^{-1}\mathbf{b}. \qquad (2)$$

Using Eq. (2), the time complexity of computing $\mathbf{x}^*$ is $O(n^2p + n^3)$, which is $O(n^2p)$ when $p \gg n$. This approach is widely applied in practice for the $p > n$ cases. However, for large dataset with high dimensional features, i.e. $p \gg n \gg 1$, this approach can be prohibitively slow.

Our contributions. In this paper, we present the first algorithm for ridge regression that runs in $o(n^2p)$ time with a provable relative-error approximation bound on the output vector. Specifically, our proposed algorithm outputs a vector $\tilde{\mathbf{x}}$ such that $\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_2 \le \epsilon \|\mathbf{x}^*\|_2$ with high probability without any assumptions on input $\mathbf{A}$ and $\mathbf{b}$. We show that our proposed algorithm runs in $O(\mathrm{nnz}(\mathbf{A}) + n^2 r \log(\frac{r}{\epsilon})/\epsilon^2)$ time, where $\mathrm{nnz}(\mathbf{A})$ is the number of non-zero entries of $\mathbf{A}$ and r is the rank of $\mathbf{A}$. Since $\mathrm{nnz}(\mathbf{A}) \le np \ll n^2p$ and $r \le n$, our algorithm is substantially faster than the existing approach, which uses $O(n^2p + n^3)$ for $p \gg n$ instances, even if $\mathbf{A}$ is full rank (in this case, our algorithm runs in $O(\mathrm{nnz}(\mathbf{A}) + n^3 \log(\frac{n}{\epsilon})/\epsilon^2)$ time).

For supplements to our main result, we also prove a risk inflation bound of our algorithm under standard statistical assumptions. In addition, we apply our techniques to several generalizations of ridge regression. In particular, we extend our algorithm to multiple ridge regression problem, where there are multiple target vectors. Similar to our result for standard ridge regression, we also prove a relative-error bound on the output. Moreover, building upon the recent results [3], we extend our techniques and design a fast relative-error approximation algorithm for a non-linear ridge regression problem.

We evaluate our algorithm on both synthetic and real datasets. The experimental results support our theoretical

[*] The first two authors contributed equally.

analysis and demonstrate that the proposed algorithm can achieve considerable speed gains and produce highly accurate outputs.

1.1 Related work

Oblivious subspace embedding (OSE). Sarlós [28] pioneered the use of OSEs for accelerating approximation algorithms for numerical linear algebra problems, including matrix multiplication, linear regression and low rank approximation. His algorithms were later improved by several work [12, 24] using different OSEs such as sparse embedding. More recent work showed that OSEs can be used to speed up other problems as well, including k-means clustering [10], approximating leverage scores [23], canonical correlation analysis [2], support vector machine [25], SVD-truncated linear regression [9], and non-linear regression [3]. In this line of research, approximation algorithms of linear regression [28, 12, 24, 27] are the most relevant to ours. However, all of these work focused on $n \gg p$ instances and the fastest algorithm [24] among them runs in $O(\mathrm{nnz}(\mathbf{A}) + p^3 \log(p)/\epsilon^2)$ time, which is inefficient in $p \gg n$ cases. Moreover, they used the sketched matrix $\mathbf{SA}$ which reduces the sample size n while we use $\mathbf{AS}^T$ which reduces feature dimension p. This distinction leads to a very different design and analysis of algorithms.

Ridge regression. The bottlenecks of solving ridge regression are constructing and inverting the kernel matrix $\mathbf{AA}^T$. In the mean time, fast kernel approximation algorithms for large datasets have been a research focus for many years. Many approximation schemes are highly successful such as low rank approximation [33], Nyström methods [7, 15, 19, 34], sampling [20, 1], incomplete Cholesky factorization [5, 16] and specialized techniques for certain classes of kernels [26, 21]. We notice that many of these proposals focused on speeding up the $n \gg p$ cases but did not necessarily improve the $p \gg n$ cases. More importantly, from these work, it is not clear how the error of kernel approximation impacts on the accuracy of the approximation result of ridge regression. This problem was recently studied in several work [13, 4, 35] under different settings. However, none of these work provided a relative-error approximation guarantee on the output vector.

The work most closely related to our results is [22]. They proposed an approximation algorithm for ridge regression by accelerating the computation of kernel matrix $\mathbf{AA}^T$ using subsampled randomized Hadamard transformation (SRHT). Their algorithm runs in $O(np\log(n)/\epsilon^2 + n^3)$ time, which is $o(n^2p)$ time. However, their algorithm does not have a provable guarantee on the error of the output vector. In addition, the risk inflation bound they proved might not hold since their proof is based on a problematic claim. We have included a detailed argument and counterexample for their proof in our supplementary material (see Section E).

2 PRELIMINARIES

2.1 NOTATION

Let $[k]$ denote the set of integers $\{1, 2, \ldots, k\}$. Given a matrix $\mathbf{M} \in \mathbb{R}^{n\times p}$ of rank r. For $i \in [n]$, let $\mathbf{M}_{(i)}$ denote the i-th row of $\mathbf{M}$ as a row vector. Let $\mathrm{nnz}(\mathbf{M})$ denote the number of non-zero entries of $\mathbf{M}$. Let $\|\mathbf{M}\|_F$ denote the Frobenius norm of $\mathbf{M}$ and let $\|\mathbf{M}\|_2$ denote the spectral norm of $\mathbf{M}$. Let $\sigma_i(\mathbf{M})$ denote the i-th largest singular value of $\mathbf{M}$ and let $\sigma_{\max}(\mathbf{M})$ and $\sigma_{\min}(\mathbf{M})$ denote the largest and smallest singular values of $\mathbf{M}$. The thin SVD of $\mathbf{M}$ is $\mathbf{M} = \mathbf{U}_M\mathbf{\Sigma}_M\mathbf{V}_M^T$, where $\mathbf{U}_M \in \mathbb{R}^{n\times r}$, $\mathbf{\Sigma}_M \in \mathbb{R}^{r\times r}$ and $\mathbf{V}_M \in \mathbb{R}^{p\times r}$.

The Moore-Penrose pseudoinverse of $\mathbf{M}$ is a $p \times n$ matrix defined by $\mathbf{M}^\dagger = \mathbf{V}_M\mathbf{\Sigma}_M^{-1}\mathbf{U}_M^T$, which can be computed in $O(n^2p)$ time when $p > n$. Finally, let $\mathbf{I}_n$ denote the $n \times n$ identity matrix and let $\mathbf{0}_n$ denote the $n \times n$ zero matrix.

2.2 OBLIVIOUS SUBSPACE EMBEDDING

We start by reviewing the definition of oblivious subspace embedding (OSE).

Definition 1. *Given any $r > 0$, $\delta \in (0, 1)$ and $\epsilon \in (0, 1)$, we call a $t \times p$ random matrix $\mathbf{S}$ an (r, δ, ϵ)-OSE, if, for any rank r matrix $\mathbf{M} \in \mathbb{R}^{p\times m}$, the following holds simultaneously for all $\mathbf{z} \in \mathbb{R}^m$,*

$$(1-\epsilon)\|\mathbf{Mz}\|_2 \le \|\mathbf{SMz}\|_2 \le (1+\epsilon)\|\mathbf{Mz}\|_2,$$

with probability at least $1 - \delta$.

In fact, many random matrices, which are widely used in machine learning, have been shown to be OSEs, e.g. Gaussian matrices [14] and random sign matrices [28]. However, many of these matrices are dense. For a dense OSE $\mathbf{S} \in \mathbb{R}^{t\times p}$, computing sketched matrix $\mathbf{SM}$ given $\mathbf{M} \in \mathbb{R}^{p\times m}$ requires time $O(t \cdot \mathrm{nnz}(\mathbf{M}))$. To speed up the computation of the sketched matrix $\mathbf{SM}$, several work sought $\mathbf{S}$ supporting fast matrix-vector multiplication [12, 32, 24]. We refer interested readers to [8] for an overview of the development of fast OSEs.

In this paper, we use a combination of two types of fast OSEs: *sparse embedding* and *subsampled randomized Hadamard transformation* (SRHT). In the following, we review their definitions and key properties.

Sparse embedding. A sparse embedding matrix $\mathbf{\Phi}_{\text{sparse}}$ is a very sparse matrix such that there is only one non-zero element per column. $\mathbf{\Phi}_{\text{sparse}} \in \mathbb{R}^{t\times p}$ can be constructed as follows. Let $h : [p] \to [t]$ be a random mapping such that for each $i \in [p]$, $h(i)$ is uniformly distributed over $[t]$. Let $\mathbf{\Phi} \in \{0, 1\}^{t\times p}$ be a binary matrix with $\mathbf{\Phi}_{h(i),i} = 1$ for each $i \in [p]$ and all remaining entries 0. Let $\mathbf{D}$ be an $p \times p$ random diagonal matrix where each diagonal entry is independent chosen $+1$ or -1 with equal probability. Finally, the sparse embedding matrix $\mathbf{\Phi}_{\text{sparse}}$ is the product of $\mathbf{D}$ and $\mathbf{\Phi}$, i.e. $\mathbf{\Phi}_{\text{sparse}} = \mathbf{\Phi}\mathbf{D}$. It is easy to see that computing $\mathbf{\Phi}_{\text{sparse}}\mathbf{M}$ takes $O(\text{nnz}(\mathbf{M}))$ time due to the sparsity of $\mathbf{\Phi}_{\text{sparse}}$.

Recently, Clarkson and Woodruff [12] showed that $\mathbf{\Phi}_{\text{sparse}}$ is an (r, δ, ϵ)-OSE if $t \geq O(r^2/\epsilon^4)$. Later, the bound on t was improved to $t \geq O(r^2/\epsilon^2)$ by Nelson and Nguyen [24]. Their result is restated in the following.

Theorem 1. [24, Theorem 3] *Given $\epsilon \in (0, 1)$, $\delta \in (0, 1)$ and $r > 0$, if $t \geq \delta^{-1}(r^2 + r)/(2\epsilon - \epsilon^2)^2$, then sparse embedding matrix $\mathbf{\Phi}_{\textit{sparse}} \in \mathbb{R}^{t \times p}$ is an (r, δ, ϵ)-OSE.*

SRHT. An SRHT matrix $\mathbf{\Phi}_{\text{srht}}$ is a highly structured matrix which allows fast, FFT-style matrix-vector multiplication. The definition of SRHT matrix $\mathbf{\Phi}_{\text{srht}} \in \mathbb{R}^{t \times p}$ is as follows. Without loss of generality, suppose that p is a power of 2 (otherwise we can pad a sufficient number of zeros). Then, $\mathbf{\Phi}_{\text{srht}}$ is given by

$$\mathbf{\Phi}_{\text{srht}} = \sqrt{\frac{p}{t}}\mathbf{R}\mathbf{H}\mathbf{D},$$

where $\mathbf{D} \in \mathbb{R}^{p \times p}$ is a random diagonal matrix whose entries are $+1$ or -1 with equal probability; $\mathbf{R} \in \mathbb{R}^{t \times p}$ are t rows from the $p \times p$ identity matrix, where the rows are chosen uniformly at random without replacement; and $\mathbf{H} \in \mathbb{R}^{p \times p}$ is a normalized Walsh-Hadamard matrix, which is defined as

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{H}_{k/2} & \mathbf{H}_{k/2} \\ \mathbf{H}_{k/2} & -\mathbf{H}_{k/2} \end{bmatrix} \text{ with } \mathbf{H}_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix},$$

and $\mathbf{H} = p^{-\frac{1}{2}}\mathbf{H}_p \in \mathbb{R}^{p \times p}$.

Using FFT-style algorithms, the product $\mathbf{\Phi}_{\text{srht}}\mathbf{M}$ can be computed in $O(np\log(p))$ time [32]. Tropp [32] showed that $\mathbf{\Phi}_{\text{srht}}$ is an OSE if $t \geq O([\sqrt{r} + \sqrt{\log(p)}]^2 \log(r)/\epsilon)$, or $t \geq O(r\log(r)/\epsilon)$ when $r > \log(p)$. His result is restated in the next theorem.

Theorem 2. [32, Lemma 4.1] *Given $\epsilon \in (0, 1)$, $\delta \in (0, 1)$ and $r > 0$, if $t \geq 6\epsilon^{-1}[\sqrt{r} + \sqrt{8\log(3p/\delta)}]^2 \log(3r/\delta)$, then SRHT matrix $\mathbf{\Phi}_{\textit{srht}} \in \mathbb{R}^{t \times p}$ is an (r, δ, ϵ)-OSE.*

2.3 COMBINATION OF SPARSE EMBEDDING AND SRHT

Sparse embedding matrix is an extremely fast OSE since computing $\mathbf{\Phi}_{\text{sparse}}\mathbf{M}$ takes only $O(\text{nnz}(\mathbf{M}))$ time, which equals to the complexity of reading $\mathbf{M}$. Meanwhile SRHT produces a sketch with $t = O(r\log(r)/\epsilon)$ rows which is smaller than sparse embedding, which requires $t = O(r^2/\epsilon^2)$. In this paper, we use the combination of SRHT and sparse embedding that enjoys the benefits from both of them. Specifically, we consider the product $\mathbf{S} = \mathbf{\Phi}_{\text{srht}}\mathbf{\Phi}_{\text{sparse}}$, where $\mathbf{\Phi}_{\text{srht}}$ is a $t \times t'$ SRHT matrix and $\mathbf{\Phi}_{\text{sparse}}$ is a $t' \times p$ sparse embedding matrix. The next theorem shows that, if $t = O([\sqrt{r} + \sqrt{\log(p)}]^2 \log(r)/\epsilon)$ and $t' = O(r^2/\epsilon^2)$, the product $\mathbf{S}$ is an OSE.

Theorem 3. *Given $\epsilon \in (0, 1)$, $\delta \in (0, 1)$ and $r > 0$, select integers $t' \geq 2\delta^{-1}(r^2 + r)/(2\epsilon/3 - \epsilon^2/9)^2$ and $t \geq 18\epsilon^{-1}[\sqrt{r} + \sqrt{8\log(6p/\delta)}]^2 \log(6r/\delta)$. Let $\mathbf{\Phi}_{\textit{sparse}}$ be a $t' \times p$ sparse embedding matrix and let $\mathbf{\Phi}_{\textit{srht}}$ be a $t \times t'$ SRHT matrix. Then the product $\mathbf{S} = \mathbf{\Phi}_{\textit{srht}}\mathbf{\Phi}_{\textit{sparse}}$ is an (r, δ, ϵ)-OSE.*

Hence, when $r \geq O(\log(p))$, the product $\mathbf{S}$ has $t = O(r\log(r)/\epsilon)$ rows, which is smaller than only using sparse embedding matrix, and computing a sketched matrix $\mathbf{SM}$ given $\mathbf{M} \in \mathbb{R}^{p \times m}$ takes $O(\text{nnz}(\mathbf{M}) + mr^2\log(r)/\epsilon^2)$ time. The proof of Theorem 3 is deferred to the supplementary material.

3 ALGORITHMS AND MAIN RESULTS

In this section, we present our approximation algorithm for ridge regression (Algorithm 1). Then, we state our main result on the approximation guarantee of our algorithm (Theorem 4). In Section 3.1, we outline the proofs of our main result.

Algorithm. Algorithm 1 takes inputs of the design matrix $\mathbf{A} \in \mathbb{R}^{n \times p}$, target vector $\mathbf{b} \in \mathbb{R}^n$, regularization parameter $\lambda > 0$ and integer parameters t' and t. The first part of Algorithm 1 is to compute the sketched matrix $\mathbf{AS}^T$, where $\mathbf{S} \in \mathbb{R}^{t \times p}$ is chosen to be the product of sparse embedding matrix $\mathbf{\Phi}_{\text{sparse}} \in \mathbb{R}^{t' \times p}$ and SRHT matrix $\mathbf{\Phi}_{\text{srht}} \in \mathbb{R}^{t \times t'}$, i.e. $\mathbf{S} = \mathbf{\Phi}_{\text{srht}}\mathbf{\Phi}_{\text{sparse}}$. As the first step, the algorithm constructs the sparse embedding matrix $\mathbf{\Phi}_{\text{sparse}}$ and the SRHT matrix $\mathbf{\Phi}_{\text{srht}}$. After that, the algorithm applies $\mathbf{\Phi}_{\text{sparse}}$ and $\mathbf{\Phi}_{\text{srht}}$ to each row $\mathbf{A}_{(i)}$ and obtains the sketched row $(\mathbf{AS}^T)_{(i)} = (\mathbf{A}_{(i)}\mathbf{\Phi}_{\text{sparse}}^T)\mathbf{\Phi}_{\text{srht}}^T$ for all $i \in [n]$. This step can be done in one pass through the rows of $\mathbf{A}$ in arbitrary order. The algorithm then combines the sketched rows $\{(\mathbf{AS}^T)_{(i)}\}_{i \in [n]}$ to form the sketched matrix $\mathbf{AS}^T$.

Next, the algorithm uses the sketched matrix $\mathbf{AS}^T$ to compute the approximate solution $\tilde{\mathbf{x}}$ of ridge regression Eq. (1). In this step, we use the following key estimation of $\tilde{\mathbf{x}}$,

$$\tilde{\mathbf{x}} = \mathbf{A}^T(\mathbf{AS}^T)^{\dagger^T}(\lambda(\mathbf{AS}^T)^{\dagger^T} + \mathbf{AS}^T)^{\dagger}\mathbf{b}. \tag{3}$$

This step requires access to $\mathbf{AS}^T$, which is computed in the previous step, and a second pass through $\mathbf{A}$ (for premultiplying $\mathbf{A}^T$). We summarize the above procedure of computing $\mathbf{AS}^T$ and $\tilde{\mathbf{x}}$ in Algorithm 1.

Main result. Our main result is the following theorem which states that, with high probability, the output $\tilde{\mathbf{x}}$ obtained in Algorithm 1 is a relative-error approximation to the optimal solution $\mathbf{x}^*$ of ridge regression .

Theorem 4. *Suppose that we are given a design matrix $\mathbf{A} \in \mathbb{R}^{n \times p}$ of rank r, a target vector $\mathbf{b} \in \mathbb{R}^n$, a regularization parameter $\lambda > 0$, accuracy parameters $\epsilon \in (0, 1)$ and $\delta \in (0, 1)$. Select integers t', t such that $t' \geq 2\delta^{-1}(r^2 + r)/(\epsilon/6 - \epsilon^2/144)^2$ and $t \geq 72\epsilon^{-1}[\sqrt{r} + \sqrt{8\log(6p/\delta)}]^2 \log(6r/\delta)$. Run Algorithm 1 with inputs*

Algorithm 1 Fast relative-error approximation algorithm of ridge regression

```
Input: design matrix $\mathbf{A} \in \mathbb{R}^{n \times p}$ ($n$ samples with $p$ features), target vector $\mathbf{b} \in \mathbb{R}^n$, regularization parameter $\lambda > 0$, integer parameters $t'$ and $t$.
Output: approximate solution $\tilde{\mathbf{x}} \in \mathbb{R}^p$ to ridge regression problem Eq. (1).
Construct sparse embedding matrix $\mathbf{\Phi}_{\text{sparse}} \in \mathbb{R}^{t' \times p}$.
Construct SRHT matrix $\mathbf{\Phi}_{\text{srht}} \in \mathbb{R}^{t \times t'}$.
for each row $\mathbf{A}_{(i)}$ of $\mathbf{A}$ in arbitrary order do
    Compute $(\mathbf{AS}^T)_{(i)} \leftarrow (\mathbf{A}_{(i)}\mathbf{\Phi}_{\text{sparse}}^T)\mathbf{\Phi}_{\text{srht}}^T$
end for
Construct $\mathbf{AS}^T$ by concatenating row vectors $\{(\mathbf{AS}^T)_{(i)}\}_{i \in [n]}$.
Compute the pseudoinverse $(\mathbf{AS}^T)^\dagger$
Set $\tilde{\mathbf{x}} \leftarrow \mathbf{A}^T(\mathbf{AS}^T)^{\dagger T}(\lambda(\mathbf{AS}^T)^{\dagger T} + \mathbf{AS}^T)^\dagger\mathbf{b}$
return $\tilde{\mathbf{x}}$
```

$\mathbf{A}$, $\mathbf{b}$, λ, t', t *and let* $\tilde{\mathbf{x}}$ *denote the output of the algorithm. Then, with probability at least* $1 - \delta$, *we have*

$$\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_2 \leq \epsilon \|\mathbf{x}^*\|_2 , \tag{4}$$

where $\mathbf{x}^*$ *is the optimal solution of ridge regression in Eq.* (1).

In addition, if $t' = O(r^2/\epsilon^2)$ *and* $t = O(r \log(r)/\epsilon)$, *the time complexity of Algorithm 1 is*

$$O\left(\text{nnz}(\mathbf{A}) + nr^2 \log\left(\frac{r}{\epsilon}\right)/\epsilon^2 + n^2 r \log(r)/\epsilon\right).$$

Running times. Set $t' = O(r^2/\epsilon^2)$ and $t = O(r \log(r)/\epsilon)$ according to Theorem 4. Then, the time complexity of each step of Algorithm 1 can be analyzed as follows. Constructing sparse embedding matrix $\mathbf{\Phi}_{\text{sparse}}$ and SRHT matrix $\mathbf{\Phi}_{\text{srht}}$ uses $O(p)$ time. Right-multiplying the sparse embedding matrix $\mathbf{A\Phi}_{\text{sparse}}^T$ takes $O(\text{nnz}(\mathbf{A}))$ time. Computing SRHT $(\mathbf{A\Phi}_{\text{sparse}}^T)\mathbf{\Phi}_{\text{srht}}^T$ uses $O(nt' \log(t')) = O(nr^2 \log(\frac{r}{\epsilon})/\epsilon^2)$ time. The pseudoinverse of $\mathbf{AS}^T$ and $\lambda(\mathbf{AS}^T)^{\dagger T} + \mathbf{AS}^T$ can be computed in $O(n^2 t) = O(n^2 r \log(r)/\epsilon)$ time. Computing the product $(\mathbf{AS}^T)^{\dagger T}(\lambda(\mathbf{AS}^T)^{\dagger T} + \mathbf{AS}^T)^\dagger\mathbf{b}$ also takes $O(n^2 t) = O(n^2 r \log(r)/\epsilon)$ time. Finally, left-multiplying $\mathbf{A}^T$ uses $O(\text{nnz}(\mathbf{A}))$ time. So, the total running time is the sum of all these operations, which is $O(\text{nnz}(\mathbf{A}) + nr^2 \log(\frac{r}{\epsilon})/\epsilon^2 + n^2 r \log(r)/\epsilon)$.

Remarks. In practice, one may not have prior knowledge on the rank r of $\mathbf{A}$. By Theorem 4, it is safe to assume that $r = n$, which is the largest possible value of r, and hence set $t' = O(n^2/\epsilon^2)$ and $t = O(n \log(n)/\epsilon)$. In this case, the running time of Algorithm 1 is $O(\text{nnz}(\mathbf{A}) + n^3 \log(\frac{n}{\epsilon})/\epsilon^2)$. This is still an $o(n^2 p)$ algorithm and is substantially faster than the standard $O(n^2 p + n^3)$ solver for $p \gg n$ instances.

In addition, the estimation method of $\tilde{\mathbf{x}}$ as in Eq. (3) holds for general OSEs $\mathbf{S}$, not necessarily limiting to the one used in Algorithm 1, i.e. $\mathbf{S} = \mathbf{\Phi}_{\text{srht}}\mathbf{\Phi}_{\text{sparse}}$. This fact is formalized in the following lemma.

Lemma 1. *Given* $\mathbf{A} \in \mathbb{R}^{n \times p}$ *of rank* r, $\mathbf{b} \in \mathbb{R}^n$, $\lambda > 0$. *Suppose that* $\mathbf{S} \in \mathbb{R}^{t \times p}$ *is an* $(r, \delta, \epsilon/4)$*-OSE for* $\epsilon \in (0, 1)$ *and* $\delta \in (0, 1)$. *Then, with probability at least* $1 - \delta$, *the approximation solution* $\tilde{\mathbf{x}}$ *obtained by Eq.* (3) *satisfies*

$$\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_2 \leq \epsilon \|\mathbf{x}^*\|_2 ,$$

where $\mathbf{x}^*$ *is the optimal solution to ridge regression Eq.* (1).

Our choice of OSE $\mathbf{S} = \mathbf{\Phi}_{\text{srht}}\mathbf{\Phi}_{\text{sparse}}$ in Algorithm 1 guarantees that the sketched matrix $\mathbf{AS}^T$ can be computed efficiently while has a small number of columns. By Lemma 1, one may use other OSEs as well, for example, SRHT $\mathbf{S} = \mathbf{\Phi}_{\text{srht}}$. This would lead to a total time complexity of $O(np \log(p) + n^2 r \log(r)/\epsilon)$, which is slower than our choice in Algorithm 1 if $\mathbf{A}$ is a sparse matrix.

3.1 PROOF

From this point on, we denote the thin SVD of matrix $\mathbf{A} \in \mathbb{R}^{n \times p}$ with rank r by $\mathbf{A} = \mathbf{U\Sigma V}^T$, with $\mathbf{U} \in \mathbb{R}^{n \times r}$, $\mathbf{\Sigma} \in \mathbb{R}^{r \times r}$ and $\mathbf{V} \in \mathbb{R}^{p \times r}$. We denote the full SVD of $\mathbf{SV}$ by $\mathbf{SV} = \mathbf{U}_\phi\mathbf{\Sigma}_\phi\mathbf{V}_\phi^T$, with $\mathbf{U}_\phi \in \mathbb{R}^{t \times r}$, $\mathbf{\Sigma}_\phi \in \mathbb{R}^{r \times r}$ and $\mathbf{V}_\phi \in \mathbb{R}^{r \times r}$. Notice that $\mathbf{V}_\phi$ is an $r \times r$ unitary matrix and therefore $\mathbf{V}_\phi\mathbf{V}_\phi^T = \mathbf{I}_r$. We will frequently use the following property of the pseudoinverse of matrix product.

Fact 1. *For any matrix* $\mathbf{A} \in \mathbb{R}^{m \times n}$ *and* $\mathbf{B} \in \mathbb{R}^{n \times p}$, *we have* $(\mathbf{AB})^\dagger = \mathbf{B}^\dagger\mathbf{A}^\dagger$, *if at least one of the following holds.*

1. $\mathbf{A}$ *has orthonormal columns.*
2. $\mathbf{B}$ *has orthonormal rows.*
3. $\mathbf{A}$ *has full column rank and* $\mathbf{B}$ *has full row rank.*

By Fact 1, we immediate obtain the following lemma.

Lemma 2. *Suppose that* $\mathbf{SV}$ *is full rank, then the pseudoinverse of* $\mathbf{AS}^T$ *is given by*

$$(\mathbf{AS}^T)^\dagger = (\mathbf{SV})^{\dagger T}\mathbf{\Sigma}^{-1}\mathbf{U}^T.$$

The first step of our proof is to represent $\mathbf{x}^*$ and $\tilde{\mathbf{x}}$ in a form that is easier to work with.

Lemma 3. *Let the optimal solution of ridge regression* $\mathbf{x}^*$ *be defined as in Eq.* (2). *We have*

$$\mathbf{x}^* = \mathbf{VG}^{-1}\mathbf{U}^T\mathbf{b},$$

where $\mathbf{G} = \lambda\mathbf{\Sigma}^{-1} + \mathbf{\Sigma}$.

Proof. Consider the *full* SVD of $\mathbf{A}$ as $\mathbf{A} = \mathbf{U}_+\mathbf{\Sigma}_+\mathbf{V}_+^T$, with $\mathbf{U}_+ \in \mathbb{R}^{n \times n}$, $\mathbf{\Sigma}_+ \in \mathbb{R}^{n \times n}$ and $\mathbf{V}_+ \in \mathbb{R}^{p \times n}$. By the relationship between thin SVD and full SVD, we can see that $\mathbf{U}_+ = [\mathbf{U}, \mathbf{U}_-]$, $\mathbf{\Sigma}_+ = \begin{bmatrix} \mathbf{\Sigma} & \\ & \mathbf{0}_{n-r} \end{bmatrix}$ and $\mathbf{V}_+ = [\mathbf{V}, \mathbf{V}_-]$, with $\mathbf{U}_- \in \mathbb{R}^{n \times (n-r)}$ and $\mathbf{V}_- \in \mathbb{R}^{p \times (n-r)}$ being column orthonormal matrices.

Now, by definition of $\mathbf{x}^*$, we have

$$\begin{aligned}\mathbf{x}^* &= \mathbf{A}^T(\mathbf{A}\mathbf{A}^T + \lambda\mathbf{I})^{-1}\mathbf{b}\\ &= \mathbf{V}_+\mathbf{\Sigma}_+\mathbf{U}_+^T(\mathbf{U}_+\mathbf{\Sigma}_+^2\mathbf{U}_+^T + \lambda\mathbf{U}_+\mathbf{U}_+^T)^{-1}\mathbf{b}\\ &= \mathbf{V}_+\mathbf{\Sigma}_+(\mathbf{\Sigma}_+^2 + \lambda\mathbf{I})^{-1}\mathbf{U}_+^T\mathbf{b}\\ &= \mathbf{V}_+\begin{bmatrix}\mathbf{\Sigma} & \\ & \mathbf{0}_{n-r}\end{bmatrix}\begin{bmatrix}(\mathbf{\Sigma}^2 + \lambda\mathbf{I}_r)^{-1} & \\ & \lambda^{-1}\mathbf{I}_{n-r}\end{bmatrix}\mathbf{U}_+^T\mathbf{b}\\ &= \mathbf{V}_+\begin{bmatrix}(\mathbf{\Sigma} + \lambda\mathbf{\Sigma}^{-1})^{-1} & \\ & \mathbf{0}_{n-r}\end{bmatrix}\mathbf{U}_+^T\mathbf{b}\\ &= \mathbf{V}(\mathbf{\Sigma} + \lambda\mathbf{\Sigma}^{-1})^{-1}\mathbf{U}^T\mathbf{b}.\end{aligned}$$

□

Lemma 4. *Define matrix $\tilde{\mathbf{G}} = \lambda\mathbf{\Sigma}^{-1} + \mathbf{\Sigma}(\mathbf{S}\mathbf{V})^T(\mathbf{S}\mathbf{V})$. Let $\tilde{\mathbf{x}}$ be defined as in Eq.* (3). *Suppose that $\mathbf{S}\mathbf{V}$ is full rank. Then, we have that $\tilde{\mathbf{G}}$ is full rank and that*

$$\tilde{\mathbf{x}} = \mathbf{V}\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{b}.$$

Proof. By the construction of $\tilde{\mathbf{x}}$, we have

$$\begin{aligned}\tilde{\mathbf{x}} &= \mathbf{A}^T\left((\mathbf{A}\mathbf{S}^T)^\dagger\right)^T\left(\lambda\left((\mathbf{A}\mathbf{S}^T)^\dagger\right)^T + \mathbf{A}\mathbf{S}^T\right)^\dagger\mathbf{b}\\ &= \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}^{-1}(\mathbf{S}\mathbf{V})^\dagger\left(\lambda\mathbf{U}\mathbf{\Sigma}^{-1}(\mathbf{S}\mathbf{V})^\dagger + \mathbf{U}\mathbf{\Sigma}(\mathbf{S}\mathbf{V})^T\right)^\dagger\mathbf{b}\\ &= \mathbf{V}(\mathbf{S}\mathbf{V})^\dagger\left(\lambda\mathbf{U}\mathbf{\Sigma}^{-1}(\mathbf{S}\mathbf{V})^\dagger + \mathbf{U}\mathbf{\Sigma}(\mathbf{S}\mathbf{V})^T\right)^\dagger\mathbf{b}\\ &= \mathbf{V}\mathbf{V}_\phi\mathbf{\Sigma}_\phi^{-1}\mathbf{U}_\phi^T\left(\lambda\mathbf{U}\mathbf{\Sigma}^{-1}\mathbf{V}_\phi\mathbf{\Sigma}_\phi^{-1}\mathbf{U}_\phi^T + \mathbf{U}\mathbf{\Sigma}\mathbf{V}_\phi\mathbf{\Sigma}_\phi\mathbf{U}_\phi^T\right)^\dagger\mathbf{b}\\ &= \mathbf{V}\mathbf{V}_\phi\mathbf{\Sigma}_\phi^{-1}\mathbf{U}_\phi^T\mathbf{U}_\phi\left(\lambda\mathbf{\Sigma}^{-1}\mathbf{V}_\phi\mathbf{\Sigma}_\phi^{-1} + \mathbf{\Sigma}\mathbf{V}_\phi\mathbf{\Sigma}_\phi\right)^\dagger\mathbf{U}^T\mathbf{b}\\ &= \mathbf{V}\mathbf{V}_\phi\mathbf{\Sigma}_\phi^{-1}\left(\lambda\mathbf{\Sigma}^{-1}\mathbf{V}_\phi\mathbf{\Sigma}_\phi^{-1} + \mathbf{\Sigma}\mathbf{V}_\phi\mathbf{\Sigma}_\phi\right)^\dagger\mathbf{U}^T\mathbf{b}, \quad (5)\end{aligned}$$

where we have repeatedly used Fact 1 and Lemma 2.

Define $\mathbf{T}_1 = \lambda\mathbf{\Sigma}^{-1}\mathbf{V}_\phi\mathbf{\Sigma}_\phi^{-1} + \mathbf{\Sigma}\mathbf{V}_\phi\mathbf{\Sigma}_\phi$. Next, we show that $\operatorname{rank}(\mathbf{T}_1) = r$. To see this, we define $\mathbf{T}_2 = \lambda\mathbf{I} + \mathbf{\Sigma}\mathbf{V}_\phi\mathbf{\Sigma}_\phi^2\mathbf{V}_\phi^T\mathbf{\Sigma}$ and notice that $\mathbf{T}_2 = \mathbf{T}_1(\mathbf{\Sigma}_\phi\mathbf{V}_\phi^T\mathbf{\Sigma})$. Since $\lambda > 0$, it is clear that $\mathbf{T}_2$ is a positive definite matrix and therefore $\operatorname{rank}(\mathbf{T}_2) = r$.

Now notice that $\operatorname{rank}(\mathbf{\Sigma}_\phi\mathbf{V}_\phi^T\mathbf{\Sigma}) = \operatorname{rank}(\mathbf{\Sigma}_\phi) = r$. Hence, we have

$$\operatorname{rank}(\mathbf{T}_1) = \operatorname{rank}(\mathbf{T}_1(\mathbf{\Sigma}_\phi\mathbf{V}_\phi^T\mathbf{\Sigma})) = \operatorname{rank}(\mathbf{T}_2) = r.$$

Then, using Fact 1 on $\mathbf{\Sigma}_\phi^\dagger$ and $\mathbf{T}_1^\dagger$, we have

$$\begin{aligned}(5) &= \mathbf{V}\mathbf{V}_\phi\left(\lambda\mathbf{\Sigma}^{-1}\mathbf{V}_\phi + \mathbf{\Sigma}\mathbf{V}_\phi\mathbf{\Sigma}_\phi^2\right)^\dagger\mathbf{U}^T\mathbf{b}\\ &= \mathbf{V}\left(\lambda\mathbf{\Sigma}^{-1} + \mathbf{\Sigma}\mathbf{V}_\phi\mathbf{\Sigma}_\phi^2\mathbf{V}_\phi^T\right)^\dagger\mathbf{U}^T\mathbf{b}\\ &= \mathbf{V}\tilde{\mathbf{G}}^\dagger\mathbf{U}^T\mathbf{b},\end{aligned}$$

where we have used Fact 1 again and that $\mathbf{S}\mathbf{V} = \mathbf{U}_\phi\mathbf{\Sigma}_\phi\mathbf{V}_\phi^T$. Finally, the rank of $\tilde{\mathbf{G}}$ is given by

$$\operatorname{rank}(\tilde{\mathbf{G}}) = \operatorname{rank}(\mathbf{T}_1\mathbf{\Sigma}_\phi\mathbf{V}_\phi^T) = \operatorname{rank}(\mathbf{T}_1) = r.$$

Hence the pseudoinverse of $\tilde{\mathbf{G}}$ equals to its inverse, i.e. $\tilde{\mathbf{G}}^\dagger = \tilde{\mathbf{G}}^{-1}$, and this concludes our proof of the lemma. □

From Lemma 3 and Lemma 4, we see that $\tilde{\mathbf{x}}$ admits a representation that is very similar to $\mathbf{x}^*$. It is clear that the key difference between $\tilde{\mathbf{x}}$ and $\mathbf{x}^*$ comes from that of $\tilde{\mathbf{G}}$ and $\mathbf{G}$.

The next lemma (Lemma 5) is our key technical lemma, which shows that $\tilde{\mathbf{G}}$ is closely related to $\mathbf{G}$ in the sense that $\mathbf{G}^{-1}$ is an approximate matrix inversion of $\tilde{\mathbf{G}}$.

Lemma 5. *Given $\epsilon \in (0, 1/4)$ and $\delta \in (0, 1)$. Let $\mathbf{S}$ be an (r, δ, ϵ)-OSE. Let $\tilde{\mathbf{G}} = \lambda\mathbf{\Sigma}^{-1} + \mathbf{\Sigma}(\mathbf{S}\mathbf{V})^T(\mathbf{S}\mathbf{V})$ and $\mathbf{G} = \lambda\mathbf{\Sigma}^{-1} + \mathbf{\Sigma}$. Notice that $\mathbf{G}$ is invertible and define $\mathbf{R} = \mathbf{G}^{-1}\tilde{\mathbf{G}} - \mathbf{I}$. Then, with probability at least $1 - \delta$, we have (a) $\mathbf{S}\mathbf{V}$ is a full rank matrix, (b) $\|\mathbf{R}\|_2 \leq 2\epsilon + \epsilon^2$, and (c)*

$$\left\|(\mathbf{I} + \mathbf{R})^{-1}\mathbf{R}\right\|_2 \leq \frac{2\epsilon + \epsilon^2}{1 - (2\epsilon + \epsilon^2)}.$$

To prove Lemma 5, we need two ingredients from linear algebra and the theory of OSEs. First, we use the following property on the stability of singular values.

Lemma 6. [31, Section 1.3.22 (iv)] *Let $\mathbf{C} \in \mathbb{R}^{m\times n}$ and $\mathbf{D} \in \mathbb{R}^{m\times n}$ be two matrices of the same size. Then, for all $i \in [\min\{m, n\}]$,*

$$|\sigma_i(\mathbf{C} + \mathbf{D}) - \sigma_i(\mathbf{C})| \leq \|\mathbf{D}\|_2\,.$$

Lemma 6 can be regarded as a generalization of Weyl's inequality to singular values of non-Hermitian matrices. We refer readers to [see 31, Section 1.3] for a proof.

The second ingredient we needed is the following characterization of OSEs.

Theorem 5. *Let $\mathbf{V} \in \mathbb{R}^{p\times r}$ be a column orthonormal matrix. Let $\mathbf{S} \in \mathbb{R}^{t\times p}$ be an (r, δ, ϵ)-OSE. Then, with probability $1 - \delta$ over the choices of $\mathbf{S}$, we have that (a) $\mathbf{S}\mathbf{V}$ is a full rank matrix and (b) for all $i \in [r]$, the i-th largest singular value of $\mathbf{S}\mathbf{V}$ is bounded by*

$$|1 - \sigma_i(\mathbf{S}\mathbf{V})| \leq \epsilon. \quad (6)$$

The proof of Lemma 6 and Theorem 5 is deferred to the supplementary material. Using them, we are now ready to prove Lemma 5.

Proof of Lemma 5. Since $\mathbf{S}$ is an (r, δ, ϵ)-OSE. By Theorem 5, we have that, with probability $1 - \delta$, $\mathbf{S}\mathbf{V}$ is a full rank matrix and all singular values of $\mathbf{S}\mathbf{V}$ are bounded in $[1 - \epsilon, 1 + \epsilon]$. In the rest of the proof, we assume this holds. And this already proves part (a) of the lemma.

We start with bounding $\|\mathbf{R}\|_2$. By the definition of $\mathbf{R}$, we have

$$\|\mathbf{R}\|_2 = \left\|\mathbf{G}^{-1}(\tilde{\mathbf{G}} - \mathbf{G})\right\|_2$$

$$
\begin{aligned}
&= \left\|(\lambda\boldsymbol{\Sigma}^{-1}+\boldsymbol{\Sigma})^{-1}\boldsymbol{\Sigma}((\mathbf{S}\mathbf{V})^T(\mathbf{S}\mathbf{V})-\mathbf{I})\right\|_2 \\
&\le \left\|(\lambda\boldsymbol{\Sigma}^{-1}+\boldsymbol{\Sigma})^{-1}\boldsymbol{\Sigma}\right\|_2 \left\|\mathbf{V}_\phi\boldsymbol{\Sigma}_\phi^2\mathbf{V}_\phi^T-\mathbf{I}\right\|_2 \\
&= \left\|(\lambda\boldsymbol{\Sigma}^{-1}+\boldsymbol{\Sigma})^{-1}\boldsymbol{\Sigma}\right\|_2 \left\|\mathbf{V}_\phi\boldsymbol{\Sigma}_\phi^2\mathbf{V}_\phi^T-\mathbf{V}_\phi\mathbf{V}_\phi^T\right\|_2 \\
&= \left\|(\lambda\boldsymbol{\Sigma}^{-1}+\boldsymbol{\Sigma})^{-1}\boldsymbol{\Sigma}\right\|_2 \left\|\boldsymbol{\Sigma}_\phi^2-\mathbf{I}\right\|_2 \\
&\le \max_i \frac{\sigma_i}{\lambda\sigma_i^{-1}+\sigma_i}((1+\epsilon)^2-1) \\
&\le 2\epsilon+\epsilon^2, \qquad (7)
\end{aligned}
$$

where we have used the fact that $\mathbf{V}_\phi$ is a unitary matrix and dropped terms that do not change spectral norm.

Now, we apply Lemma 6 by setting $\mathbf{C}=\mathbf{I}$ and $\mathbf{D}=\mathbf{R}$. Then, for all $i\in[r]$, we have

$$\sigma_i(\mathbf{I}+\mathbf{R}) \ge 1-\|\mathbf{R}\|_2 \ge 1-(2\epsilon+\epsilon^2). \qquad (8)$$

Hence, we have

$$
\begin{aligned}
\left\|(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\right\|_2 &\le \left\|(\mathbf{I}+\mathbf{R})^{-1}\right\|_2 \|\mathbf{R}\|_2 \\
&\le (\sigma_{\min}(\mathbf{I}+\mathbf{R}))^{-1}\|\mathbf{R}\|_2 \\
&\le \frac{2\epsilon+\epsilon^2}{1-(2\epsilon+\epsilon^2)}.
\end{aligned}
$$

□

We are now ready to prove our main results: Lemma 1 and Theorem 4.

Proof of Lemma 1. Let $\epsilon'=\epsilon/4$ and recall the definition $\mathbf{R}=\mathbf{G}^{-1}\tilde{\mathbf{G}}-\mathbf{I}$. Since $\mathbf{S}$ is an (r,δ,ϵ')-OSE. By Lemma 5, with probability $1-\delta$, we have that $\mathbf{SV}$ is a full rank matrix and that $\left\|(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\right\|_2 \le \frac{2\epsilon'+\epsilon'^2}{1-(2\epsilon'+\epsilon'^2)}$. In the rest of the proof, we assume that this event happens.

Since $\mathbf{SV}$ is a full rank matrix. Applying Lemma 3 and Lemma 4, we have

$$
\begin{aligned}
\|\tilde{\mathbf{x}}-\mathbf{x}^*\|_2 &= \left\|\mathbf{V}(\tilde{\mathbf{G}}^{-1}-\mathbf{G}^{-1})\mathbf{U}^T\mathbf{b}\right\|_2 \\
&= \left\|(\tilde{\mathbf{G}}^{-1}-\mathbf{G}^{-1})\mathbf{U}^T\mathbf{b}\right\|_2, \qquad (9)
\end{aligned}
$$

where we have dropped the unitary term $\mathbf{V}$ which does not change l_2 norm.

Next, we write $\tilde{\mathbf{G}}=\mathbf{G}(\mathbf{I}+\mathbf{R})$. This means that $\tilde{\mathbf{G}}^{-1}=(\mathbf{I}+\mathbf{R})^{-1}\mathbf{G}^{-1}$. Therefore,

$$
\begin{aligned}
(9) &= \left\|\left((\mathbf{I}+\mathbf{R})^{-1}-\mathbf{I}\right)\mathbf{G}^{-1}\mathbf{U}^T\mathbf{b}\right\|_2 \\
&= \left\|-(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\mathbf{G}^{-1}\mathbf{U}^T\mathbf{b}\right\|_2 \qquad (10) \\
&\le \left\|(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\right\|_2 \left\|\mathbf{G}^{-1}\mathbf{U}^T\mathbf{b}\right\|_2 \\
&= \left\|(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\right\|_2 \|\mathbf{x}^*\|_2 \\
&\le \frac{2\epsilon'+\epsilon'^2}{1-(2\epsilon'+\epsilon'^2)}\|\mathbf{x}^*\|_2 \qquad (11) \\
&\le 4\epsilon'\|\mathbf{x}^*\|_2 = \epsilon\|\mathbf{x}^*\|_2,
\end{aligned}
$$

where Eq. (10) follows from matrix inversion lemma, i.e. $\mathbf{C}^{-1}-\mathbf{D}^{-1}=-\mathbf{C}^{-1}(\mathbf{C}-\mathbf{D})\mathbf{D}^{-1}$ for any squared matrices $\mathbf{C}$ and $\mathbf{D}$ of the same size, and Eq. (11) follows from the assumption on $\left\|(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\right\|_2$. □

Proof of Theorem 4. It is easy to see that the solution $\tilde{\mathbf{x}}$ returned by Algorithm 1 is given by Eq. (3) with $\mathbf{S}=\boldsymbol{\Phi}_{\text{srht}}\boldsymbol{\Phi}_{\text{sparse}}$. Therefore, the bound on $\|\tilde{\mathbf{x}}-\mathbf{x}\|_2$ follows immediately from Lemma 1 and Theorem 3 which shows that $\mathbf{S}=\boldsymbol{\Phi}_{\text{srht}}\boldsymbol{\Phi}_{\text{sparse}}$ is an $(r,\delta,\epsilon/4)$-OSE. And the running time analysis of Algorithm 1 is given in Section 3. □

4 RISK INFLATION BOUND

In this section, we study the risk inflation of the approximate solution $\tilde{\mathbf{x}}$ returned by Algorithm 1 with respect to the optimal solution $\mathbf{x}^*$ of ridge regression. We begin with review the definition of risk of ridge regression. To properly define the risk, we need to that $\mathbf{A}$ and $\mathbf{b}$ have the linear relationship as follows

$$\mathbf{b}=\mathbf{A}\mathbf{x}_0+\mathbf{e}, \qquad (12)$$

where $\mathbf{x}_0\in\mathbb{R}^p$ is an unknown vector which is assumed to be the "true" parameter and $\mathbf{e}\in\mathbb{R}^n$ is independent noise in each coordinate, with $\mathbf{E}[e_i]=0$ and $\mathbf{Var}[e_i]=\sigma^2$. Under this assumption, the risk of any vector $\hat{\mathbf{b}}\in\mathbb{R}^n$ is given by

$$\text{risk}(\hat{\mathbf{b}}) \triangleq \frac{1}{n}\mathbf{E}\left[\left\|\hat{\mathbf{b}}-\mathbf{A}\mathbf{x}_0\right\|_2^2\right],$$

where the expectation is taken over the randomness of noise [4].

The following theorem shows that, compared with the optimal solution $\mathbf{x}^*$, the approximate solution $\tilde{\mathbf{x}}$ returned by Algorithm 1 increases the risk by a small additive factor.

Theorem 6. *Given $\mathbf{A}\in\mathbb{R}^{n\times p}$ of rank r, $\mathbf{b}\in\mathbb{R}^n$, $\lambda>0$, $\epsilon\in(0,1)$ and $\lambda\in(0,1)$. Assume that $\mathbf{A}$ and $\mathbf{b}$ have the linear relationship as in Eq. (12). Let $\tilde{\mathbf{x}}$ denote the output of Algorithm 1 with inputs $\mathbf{A}$, $\mathbf{b}$, λ, $t'=\lceil 2\delta^{-1}(r^2+r)/(\epsilon/6-\epsilon^2/144)^2\rceil$ and $t=\left\lceil 72\epsilon^{-1}[\sqrt{r}+\sqrt{8\log(6p/\delta)}]^2\log(6r/\delta)\right\rceil$. Let $\mathbf{x}^*$ denote the optimal solution of ridge regression. Then, with probability at least $1-\delta$,*

$$\text{risk}(\tilde{\mathbf{b}}) \le \text{risk}(\mathbf{b}^*)+\frac{3\epsilon}{n}\|\mathbf{A}\|_2^2\left(\|\mathbf{x}_0\|^2+\sigma^2\rho^2\right), \qquad (13)$$

where we define $\tilde{\mathbf{b}}=\mathbf{A}\tilde{\mathbf{x}}$ and $\mathbf{b}^*=\mathbf{A}\mathbf{x}^*$; we also define $\rho^2=\sum_{i\in[r]}\left(\frac{\sigma_i}{\sigma_i^2+\lambda}\right)^2$ and σ_i is the i-th largest singular value of $\mathbf{A}$.

5 EXTENSIONS

In this section, we present two extensions to our algorithm. First, we consider the multiple response ridge regression problem, and obtain an efficient approximation algorithm with relative-error guarantee similar with Algorithm 1. Second, combining with the recent results of Avron et al. [3], we present a fast relative-error approximation algorithm of a special nonlinear ridge regression problem called structured ridge regression.

5.1 MULTIPLE RESPONSE RIDGE REGRESSION

In this part, we generalize our techniques to solve multiple response ridge regression [11]. The multiple response ridge regression problem is defined as follows. Given a design matrix $\mathbf{A} \in \mathbb{R}^{n\times p}$, m target vectors (responses) $\mathbf{B} \in \mathbb{R}^{p\times m}$ and a regression parameter $\lambda > 0$, the multiple response regression problem is to find an $n \times m$ matrix $\mathbf{X}^*$ such that

$$\mathbf{X}^* = \underset{\mathbf{X}\in\mathbb{R}^{n\times m}}{\arg\min} \left\|\mathbf{A}\mathbf{X} - \mathbf{B}\right\|_F^2 + \lambda \left\|\mathbf{X}\right\|_F^2 . \qquad (14)$$

The optimal solution of Eq. (14) is given by

$$\mathbf{X}^* = \mathbf{A}^T(\lambda\mathbf{I}_n + \mathbf{A}\mathbf{A}^T)^{-1}\mathbf{B}. \qquad (15)$$

It is clear that Eq. (15) takes $O(n^2p + n^3 + nm)$ time to compute, which is expensive if $p \gg n \gg 1$.

Next, we generalize our techniques to solve multiple response regression problem. The first step is to compute the sketched matrix $\mathbf{A}\mathbf{S}^T$, where $\mathbf{S} = \mathbf{\Phi}_{\text{srht}}\mathbf{\Phi}_{\text{sparse}}$. Notice that this step is identical to that of Algorithm 1, which uses one pass through $\mathbf{A}$. Then, we use the following generalized version of Eq. (3) to compute the approximate solution $\tilde{\mathbf{X}}$,

$$\tilde{\mathbf{X}} = \mathbf{A}^T(\mathbf{A}\mathbf{S}^T)^{\dagger T}(\lambda(\mathbf{A}\mathbf{S}^T)^{\dagger T} + \mathbf{A}\mathbf{S}^T)^{\dagger}\mathbf{B}, \qquad (16)$$

which uses a second pass through $\mathbf{A}$. We show that the approximate solution $\tilde{\mathbf{X}}$ given by Eq. (16) is a relative-error approximation of $\mathbf{X}^*$ in the following theorem.

Theorem 7. *Given $\mathbf{A} \in \mathbb{R}^{n\times p}$ of rank r, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\lambda > 0$, parameter $\epsilon \in (0,1)$ and $\delta \in (0,1)$. Select integers t', t such that $t' \geq 2\delta^{-1}(r^2 + r)/(\epsilon/6 - \epsilon^2/144)^2$ and $t \geq 72\epsilon^{-1}[\sqrt{r} + \sqrt{8\log(6p/\delta)}]^2 \log(6r/\delta)$. Let $\mathbf{S} = \mathbf{\Phi}_{srht}\mathbf{\Phi}_{sparse}$, where $\mathbf{\Phi}_{sparse} \in \mathbb{R}^{t'\times p}$ is a sparse embedding matrix and $\mathbf{\Phi}_{srht} \in \mathbb{R}^{t\times t'}$ is an SRHT matrix. Then, with probability at least $1-\delta$, we have*

$$\left\|\tilde{\mathbf{X}} - \mathbf{X}^*\right\|_F \leq \epsilon \left\|\mathbf{X}^*\right\|_F ,$$

where $\tilde{\mathbf{X}}$ is given by Eq. (16) *and $\mathbf{X}^*$ is the optimal solution to multiple response ridge regression Eq.* (14). *In addition, the total time complexity of computing $\mathbf{A}\mathbf{S}^T$ and $\tilde{\mathbf{X}}$ is $O(\text{nnz}(\mathbf{A}) + nr^2\log(\frac{r}{\epsilon})/\epsilon^2 + n^2r\log(r)/\epsilon + nm)$.*

5.2 STRUCTURED RIDGE REGRESSION

In this part, we consider a non-linear ridge regression problem, called *structured ridge regression*, which is closely related to kernel ridge regression with polynomial kernels. Structured ridge regression uses a non-linear kernel expansion function φ_q, which is studied recently by Avron et al. [3] under the context of (non-regularized) structured regression. The kernel expansion function φ_q maps a p-dimensional vector $\mathbf{a}$ to a pq-dimensional vector $\varphi_q(\mathbf{a}) = \{a_i^{j-1}\}_{(i,j)\in[p]\times[q]}$ for $q > 1$. The definition of φ_q corresponds to the kernel function $k_{\varphi_q}(\mathbf{a},\mathbf{b}) = \varphi_q(\mathbf{a})^T\varphi_q(\mathbf{b}) = \sum_{(i,j)\in[p]\times[q]}(a_ib_i)^{j-1}$, for any p-dimensional $\mathbf{a}$ and $\mathbf{b}$. Clearly, k_{φ_q} is related to polynomial kernels, which is defined as $k_q(\mathbf{a},\mathbf{b}) = (\mathbf{a}^T\mathbf{b})^q = \left(\sum_{i\in[p]} a_ib_i\right)^q$. For more detailed discussion and the connections of φ_q to other kernels, we refer interested readers to [3].

In the following, we define structured ridge regression, which can be seen an l_2 regularized version of structured regression proposed by Avron et al. [3],

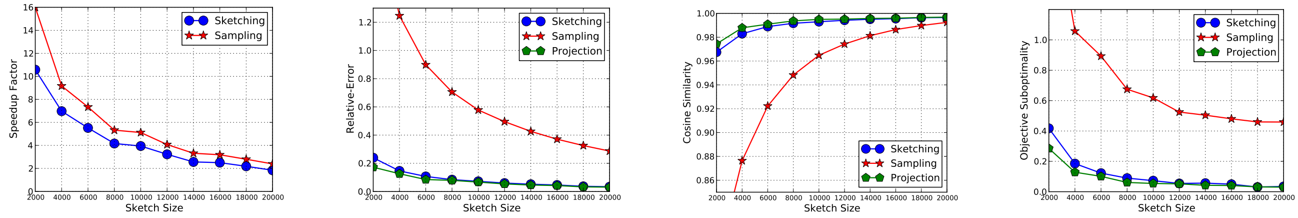
$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathbb{R}^{pq}}{\arg\min} \left\|\varphi_q(\mathbf{A})\mathbf{x} - \mathbf{b}\right\|_2^2 + \lambda \left\|\mathbf{x}\right\|_2^2 , \qquad (17)$$

where $\varphi_q(\mathbf{A})$ is an $n \times pq$ matrix consisting of n expanded samples, i.e. its i-th row vector is $\varphi_q(\mathbf{A})_{(i)} = \varphi_q(\mathbf{A}_{(i)})$ for all $i \in [n]$. Clearly, Eq (17) is a non-linear ridge regression problem. For this problem, the dual space approach gives that $\mathbf{x}^* = \varphi_q(\mathbf{A})^T(\mathbf{K} + \lambda\mathbf{I}_n)^{-1}\mathbf{b}$, where $\mathbf{K} = \varphi_q(\mathbf{A})\varphi_q(\mathbf{A})^T$. It is clear that computing $\mathbf{x}^*$ using this approach takes $O(n^2pq + n^3)$ time.

We extend our techniques to accelerate the computation of structured ridge regression for $p \gg n$ instances by using the following property of φ_q. Avron et al. [3] showed that there exists a fast multiplication algorithm which computes the product $\varphi_q(\mathbf{A})\mathbf{\Phi}_{\text{sparse}}^T$ in $O((\text{nnz}(\mathbf{A}) + nqt')\log^2(q))$ time by exploiting the structure of $\varphi_q(\mathbf{A})$. Therefore, we only need to modify Algorithm 1 to use the fast multiplication algorithm for computing the sketched matrix $(\varphi_q(\mathbf{A})\mathbf{\Phi}_{\text{sparse}}^T)\mathbf{\Phi}_{\text{srht}}^T$; then run the modified algorithm with input $\varphi_q(\mathbf{A})$ and $\mathbf{b}$. We show that this procedure gives a relative-error approximation algorithm for structured ridge regression that runs in $O\left(\text{nnz}(\mathbf{A})\log^2(q) + n^3q\log^2(q)/\epsilon^2 + n^3\log\left(\frac{n}{\epsilon}\right)/\epsilon^2\right)$ time, which is faster than the dual space approach when $p \gg n$. For constant q, this is also asymptotically faster than solving kernel ridge regression with polynomial kernel in dual space, whose computational complexity is $O(n^2p + n^3)$. We defer the detailed description of the approximation algorithm and its related analysis to the supplementary material (see Section D).

6 EXPERIMENTS

Baselines. We compare the performance of our algorithm SKETCHING (Algorithm 1) to three baselines. The first baseline is the STANDARD algorithm, which computes the optimal solution using the dual space approach in Eq. (2). The other two baselines use popular randomized dimen-



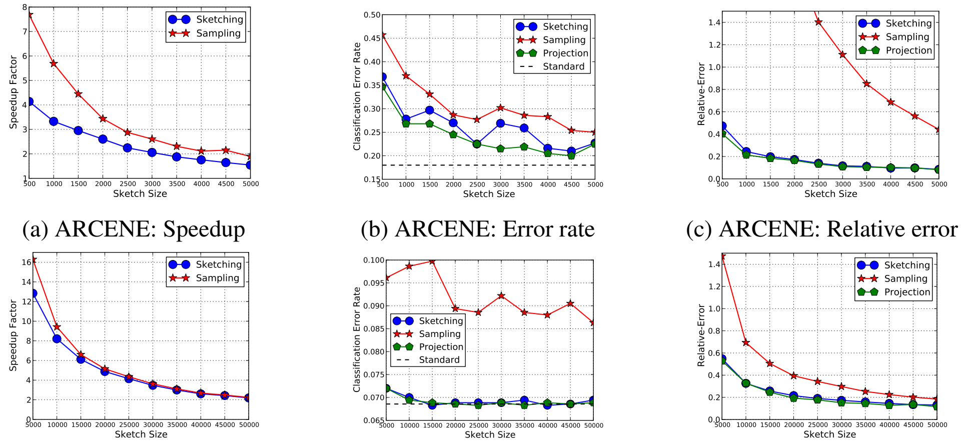
(a) Speedup

(b) Relative error

(c) Cosine similarity

(d) Objective suboptimality

Figure 1: Quality-of-approximation and running times on synthetic dataset



(a) ARCENE: Speedup

(b) ARCENE: Error rate

(c) ARCENE: Relative error

(d) DOROTHEA: Speedup

(e) DOROTHEA: Error rate

(f) DOROTHEA: Relative error

Figure 2: Classification accuracy and running times on realworld datasets

sionality reduction methods, including sampling and random projection. The SAMPLING algorithm simply samples a subset of t columns of $\mathbf{A}$ uniformly at random. The PROJECTION algorithm post-multiplies $\mathbf{A}$ by a random sign matrix $\mathbf{\Phi}_{\text{sign}}^T \in \mathbb{R}^{p\times t}$, with each entry of $\mathbf{\Phi}_{\text{sign}}$ having value chosen from $\{\pm 1/\sqrt{t}\}$ uniformly at random. Then, sketched matrices with t columns obtained by SAMPLING and PROJECTION are plugged in Eq. (3) to compute an approximate solution $\tilde{\mathbf{x}}$ of ridge regression. Finally, notice that $\mathbf{\Phi}_{\text{sign}}$ is also an OSE for sufficiently large t [28] and therefore, by Lemma 1, the PROJECTION algorithm produces a relative-error approximation as well. However, for dense $\mathbf{A}$, computing $\mathbf{A}\mathbf{\Phi}_{\text{sign}}^T$ alone takes $O(tnp)$ time, which is even slower than the STANDARD algorithm when $t > n$. Hence, we do not compare its running time to other competing algorithms.

Implementation. Our implementation of Algorithm 1 is slightly different from its description in two places. First, Algorithm 1, and its analysis, uses the Walsh-Hadamard transformation (as a step of SRHT), while our implementation uses discrete Hartley transformation (DHT) [30]. DHT has a highly optimized implementation provided in FFTW package [17]. In addition, it is possible to show that DHT or other Fourier-type transformations have a guarantee similar to Walsh-Hadamard transformation [2, 32]. Second, we set $t' = 2t$, i.e. the sketch size of sparse embedding is two times larger than that of SRHT. Empirically, this setting offers a good trade-off between accuracy and speed. All competing algorithms are implemented using C++ and the experiments are conducted on a standard workstation using a single core.

6.1 SYNTHETIC DATASET

Setup. We generate the $n \times p$ design matrix $\mathbf{A}$ using the following method, such that each row (sample) of $\mathbf{A}$ contains an s-dimensional signal and p-dimensional noises. Specifically, we define $\mathbf{A} = \mathbf{M\Sigma V}^T + \alpha\mathbf{E}$. Here, $\mathbf{M}$ is an $n \times s$ matrix which represents the signals, and each entry $M_{ij} \sim \mathcal{N}(0,1)$ is an i.i.d Gaussian random variable. $\mathbf{\Sigma}$ is an $s \times s$ diagonal matrix and the diagonal entries are given by $\Sigma_{ii} = 1 - (i-1)/p$ for each $i \in [s]$. $\mathbf{V}$ is a $p \times n$ column orthonormal matrix which contains a random s-dimensional subspace of $\mathbb{R}^p$. Notice that $\mathbf{M\Sigma V}^T$ is a rank s matrix with linearly decreasing singular values. $\mathbf{E}$ is an $n \times p$ matrix which contributes the additive Gaussian noise $E_{ij} \sim \mathcal{N}(0,1)$. $\alpha > 0$ is a parameter chosen to balance the energy of signals $\mathbf{M\Sigma V}^T$ and the energy of noises $\mathbf{E}$. In this experiment, we choose $\alpha = 0.05$ which brings $\left\|\mathbf{M\Sigma V}^T\right\|_F \approx \alpha \left\|\mathbf{E}\right\|_F$. Then, we generate the target vector $\mathbf{x} \in \mathbb{R}^p$ with $x_i \sim \mathcal{N}(0,1)$. Finally, the target vector

$\mathbf{b} \in \mathbb{R}^n$ is given by $\mathbf{b} = \mathbf{A}\mathbf{x} + \gamma\mathbf{e}$, where $e_i \sim \mathcal{N}(0,1)$ and $\gamma = 5$.

Metrics. We measure the performance of our algorithm and baselines both in terms of accuracy and speedup factor. More specifically, let $\mathbf{x}^*$ denote the optimal solution produced by the standard algorithm and let $\tilde{\mathbf{x}}$ denote the output vector returned by an approximation algorithm. To evaluate the accuracy of approximation, we compute three metrics: *relative error*: $\frac{\|\tilde{\mathbf{x}}-\mathbf{x}^*\|_2}{\|\mathbf{x}^*\|_2}$; *cosine similarity*: $\frac{\tilde{\mathbf{x}}^T\mathbf{x}*}{\|\tilde{\mathbf{x}}\|_2\|\mathbf{x}^*\|_2}$; *objective suboptimality*: $\frac{f(\tilde{\mathbf{x}})}{f(\mathbf{x}^*)} - 1$, with $f(\mathbf{x}) \triangleq \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_2^2$. In addition, the speedup factor is given by the ratio between the time used by STANDARD algorithm and that of a competing algorithm.

Results. In the experiment, we set $n = 500$, $p = 50000$ and $s = 50$. We run the competing algorithms with 10 different choices of t within range $[2000, 20000]$. The results are shown in Figure 1. Figure 1(a) reports the speed up of approximation algorithms with respect to the STANDARD algorithm. We see that our algorithm SKETCHING is slightly slower than the SAMPLING algorithm, but both of them speed up considerably with respect to the STANDARD algorithm. Figure 1(b), (c) and (d) plot the accuracy metrics of the competing algorithms. We see that indeed the accuracy of approximation improves as the sketch size t increases. In addition, both of our SKETCHING algorithm and the PROJECTION algorithm output a significantly more accurate solution than the SAMPLING algorithm. Notably, when the sketch size $t \approx 10000$, our algorithm SKETCHING has a relative-error smaller than 10%, cosine similarity larger than 99% and objective suboptimality less than 10%; meanwhile speeds up the computation about 4 times.

6.2 REALWORLD DATASETS

Setup. We also test the proposed algorithm on two binary classification datasets: ARCENE and DOROTHEA [18]. Both datasets are publicly available from the UCI repository [6]. ARCENE contains 200 samples (100 for training and 100 for testing) with 10000 real valued features. DOROTHEA consists of 1150 samples (800 for training and 350 for testing) with 100000 binary valued features. We apply ridge regression on both classification tasks by setting the responses to be $+1$ for positive examples and -1 for negative examples. We run ridge regression algorithms on the training data to compute the feature weights and measure the classification error rate on the testing data. For each dataset, we test 10 different choices of sketch size t and record the classification error rates and speed up factors of competing algorithms.

Results. The experiment results are shown in Figure 2. From the results, we observe that the classification error decreases as the sketch size t increases. It is also clear that, using the same sketch size t, SKETCHING and PROJECTION produce more accurate predictions than SAMPLING. On the other hand, SKETCHING and SAMPLING algorithms are considerably faster than the STANDARD algorithm. From the results, we see that our algorithm SKETCHING substantially speeds up the computation, while attains a very small increase in error rate. For ARCENE dataset, when $t \approx 3000$, SKETCHING accelerates the computation by 2.1 times while increases the error rate by 4.5%; and, for DOROTHEA dataset, when $t \approx 20000$, the speedup of SKETCHING is about 4.1 times and the error rate is almost the same to the STANDARD algorithm. In addition, we continue to observe that the relative-error decreases as t increases. The SKETCHING algorithm and the PROJECTION algorithm outperform the SAMPLING algorithm in terms of accuracy on both datasets. We remark that, for moderately large t, the SKETCHING algorithm achieves a relative-error that is smaller than 20% on both datasets.

7 CONCLUSIONS

We presented an efficient relative-error approximation algorithm for ridge regression for $p \gg n$ cases. Our algorithm runs in $\tilde{O}(\text{nnz}(A) + n^3/\epsilon^2)$ time, which is substantially faster than the existing $O(n^2p + n^3)$ algorithm for large p instances. In addition, we analyzed the risk inflation of our algorithm and extended our techniques to design fast relative-error approximation algorithms for multiple response ridge regression and structured ridge regression. We reported experimental results of our algorithm on both synthetic and real datasets, which supported our analysis and demonstrated good practical performance.

Acknowledgments

The work described in this paper was fully supported by the National Grand Fundamental Research 973 Program of China (No. 2014CB340401 and No. 2014CB340405), the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK 413213 and CUHK 14205214), Microsoft Research Asia Regional Seed Fund in Big Data Research (Grant No. FY13-RES-SPONSOR-036) and Research Grants Council of the Hong Kong S.A.R. (Project no. CUHK419413).

References

[1] Ahmed El Alaou and Michael W. Mahoney. Fast randomized kernel methods with statistical guarantees. *Technical Report*, 2014.

[2] Haim Avron, Christos Boutsidis, Sivan Toledo, and Anastasios Zouzias. Efficient dimensionality reduction for canonical correlation analysis. In *ICML*, 2013.

[3] Haim Avron, Vikas Sindhwani, and David Woodruff. Sketching structured matrices for faster nonlinear regression. In *NIPS*. 2013.

[4] Francis Bach. Sharp analysis of low-rank kernel matrix approximations. In *COLT*, 2013.

[5] Francis R Bach and Michael I Jordan. Kernel independent component analysis. *JMLR*, 2003.

[6] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[7] Mohamed-Ali Belabbas and Patrick J Wolfe. Spectral methods in machine learning and new strategies for very large datasets. *PNAS*, 2009.

[8] Jean Bourgain and Jelani Nelson. Toward a unified theory of sparse dimensionality reduction in euclidean space. *Technical Report*, 2013.

[9] C. Boutsidis and M. Magdon-Ismail. Faster svd-truncated regularized least-squares. In *ISIT*, 2014.

[10] Christos Boutsidis, Anastasios Zouzias, and Petros Drineas. Random projections for k-means clustering. In *NIPS*. 2010.

[11] Leo Breiman and Jerome H Friedman. Predicting multivariate responses in multiple linear regression. *J R STAT SOC B*, 1997.

[12] Kenneth L Clarkson and David P Woodruff. Low rank approximation and regression in input sparsity time. In *STOC*, 2013.

[13] Corinna Cortes, Mehryar Mohri, and Ameet Talwalkar. On the impact of kernel approximation on learning accuracy. In *AISTATS*, 2010.

[14] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *RS&A*, 2003.

[15] Petros Drineas and Michael W Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *JMLR*, 2005.

[16] Shai Fine and Katya Scheinberg. Efficient svm training using low-rank kernel representations. *JMLR*, 2002.

[17] Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *P IEEE*, 2005.

[18] Isabelle Guyon. Design of experiments of the nips 2003 variable selection benchmark. In *NIPS 2003 workshop on feature extraction and feature selection*, 2003.

[19] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. On sampling-based approximate spectral decomposition. In *ICML*, 2009.

[20] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling techniques for the nystrom method. In *AISTATS*, 2009.

[21] Quoc Le, Tamas Sarlos, and Alexander Smola. Fastfood-computing hilbert space expansions in loglinear time. In *ICML*, 2013.

[22] Yichao Lu, Paramveer Dhillon, Dean P Foster, and Lyle Ungar. Faster ridge regression via the subsampled randomized hadamard transform. In *NIPS*, 2013.

[23] Michael W Mahoney, Petros Drineas, Malik Magdon-Ismail, and David P Woodruff. Fast approximation of matrix coherence and statistical leverage. In *ICML*, 2012.

[24] Jelani Nelson and Huy L. Nguyen. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *FOCS*, 2013.

[25] Saurabh Paul, Christos Boutsidis, Malik Magdon-Ismail, and Petros Drineas. Random projections for support vector machines. In *AISTATS*, 2013.

[26] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NIPS*, 2007.

[27] Garvesh Raskutti and Michael Mahoney. Statistical and algorithmic perspectives on randomized sketching for ordinary least-squares. In *ICML*, 2015.

[28] Tamas Sarlós. Improved approximation algorithms for large matrices via random projections. In *FOCS*, 2006.

[29] Craig Saunders, Alexander Gammerman, and Volodya Vovk. Ridge regression learning algorithm in dual variables. In *ICML*, 1998.

[30] H.V. Sorensen, D.L. Jones, C.S. Burrus, and M. Heideman. On computing the discrete hartley transform. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 1985.

[31] Terence Tao. *Topics in random matrix theory*. 2012.

[32] Joel A Tropp. Improved analysis of the subsampled randomized hadamard transform. *AADA*, 2011.

[33] Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *NIPS*, 2001.

[34] Kai Zhang, Ivor W Tsang, and James T Kwok. Improved nyström low-rank approximation and error analysis. In *ICML*, 2008.

[35] Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression. In *COLT*, 2013.

Supplementary Material for "Fast Relative-Error Approximation Algorithm for Ridge Regression"

A Proof of Theorem 3

For convenience, we restate the theorem in the following.

Theorem 3. *Given $\epsilon \in (0,1)$, $\delta \in (0,1)$ and $r > 0$, select integers $t' \geq 2\delta^{-1}(r^2+r)/(2\epsilon/3-\epsilon^2/9)^2$ and $t \geq 18\epsilon^{-1}[\sqrt{r}+\sqrt{8\log(6p/\delta)}]^2\log(6r/\delta)$. Let $\mathbf{\Phi}_{sparse} \in \mathbb{R}^{t'\times p}$ be a $t' \times p$ sparse embedding matrix and let $\mathbf{\Phi}_{srht} \in \mathbb{R}^{t\times t'}$ be a $t \times t'$ SRHT matrix. Then the product $\mathbf{S} = \mathbf{\Phi}_{srht}\mathbf{\Phi}_{sparse}$ is an (r,δ,ϵ)-OSE.*

Proof. By Theorem 1, $\mathbf{\Phi}_{\text{sparse}} \in \mathbb{R}^{t'\times p}$ is an $(r,\delta/2,\epsilon/3)$-OSE. Similarly, by Theorem 2, $\mathbf{\Phi}_{\text{srht}} \in \mathbb{R}^{t'\times t}$ is also an $(r,\delta/2,\epsilon/3)$-OSE.

Now fix an arbitrary matrix $\mathbf{M} \in \mathbb{R}^{p\times m}$ of rank r. Since $\mathbf{\Phi}_{\text{sparse}}$ is an $(r,\delta/2,\epsilon/3)$-OSE, by Definition 1, we have

$$(1-\epsilon/3)\left\|\mathbf{Mz}\right\|_2 \leq \left\|\mathbf{\Phi}_{\text{sparse}}\mathbf{Mz}\right\|_2 \leq (1+\epsilon/3)\left\|\mathbf{Mz}\right\|_2, \tag{18}$$

holds for all $\mathbf{z} \in \mathbb{R}^m$ simultaneously with probability at least $1-\delta/2$. Eq. (18) also implies that $\operatorname{rank}(\mathbf{\Phi}_{\text{sparse}}\mathbf{M}) = \operatorname{rank}(\mathbf{M}) = r$. Now, conditioning on the event that Eq. (18) holds, and using the fact that $\mathbf{\Phi}_{\text{srht}}$ is an $(r,\delta/2,\epsilon/3)$-OSE, we have

$$(1-\epsilon/3)\left\|\mathbf{\Phi}_{\text{sparse}}\mathbf{Mz}\right\|_2 \leq \left\|\mathbf{\Phi}_{\text{srht}}(\mathbf{\Phi}_{\text{sparse}}\mathbf{Mz})\right\|_2 \leq (1+\epsilon/3)\left\|\mathbf{\Phi}_{\text{sparse}}\mathbf{Mz}\right\|_2, \tag{19}$$

holds for all $\mathbf{z} \in \mathbb{R}^m$ simultaneously with probability at least $1-\delta/2$. When both Eq. (18) and Eq. (19) hold, we have, for any $\mathbf{z} \in \mathbb{R}^m$,

$$\begin{aligned}\left\|\mathbf{\Phi}_{\text{srht}}(\mathbf{\Phi}_{\text{sparse}}\mathbf{Mz})\right\|_2 &\leq (1+\epsilon/3)\left\|\mathbf{\Phi}_{\text{sparse}}\mathbf{Mz}\right\|_2\\ &\leq (1+\epsilon/3)^2\left\|\mathbf{Mz}\right\|_2\\ &\leq (1+\epsilon)\left\|\mathbf{Mz}\right\|_2,\end{aligned} \tag{20}$$

where we have used Eq. (18), Eq. (19) and the fact that $\epsilon < 1$. The other direction: $\left\|\mathbf{\Phi}_{\text{srht}}(\mathbf{\Phi}_{\text{sparse}}\mathbf{Mz})\right\|_2 \geq (1-\epsilon)\left\|\mathbf{Mz}\right\|_2$ can be proved using the same method. Now, notice that, by union bound, the probability that both Eq. (18) and Eq. (19) hold simultaneously for any $\mathbf{z}$ is at least $1-\delta$. This concludes our proof. □

B Proof of Theorem 6

Proof. Let $\tilde{\mathbf{b}} = \mathbf{A}\tilde{\mathbf{x}}$ denote the prediction using approximated $\tilde{\mathbf{x}}$ returned by Algorithm 1. Let $\mathbf{b}^* = \mathbf{A}\mathbf{x}^*$ denote the prediction using optimal solution $\mathbf{x}^*$ of Eq. (1). Define $\mathbf{b}_0 = \mathbf{A}\mathbf{x}_0$. Let the thin SVD of $\mathbf{A}$ be $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$.

Using the classical bias-variance decomposition [4], for any $\hat{\mathbf{b}} \in \mathbb{R}^n$, we have

$$\operatorname{risk}(\hat{\mathbf{b}}) = \operatorname{bias}(\hat{\mathbf{b}}) + \operatorname{var}(\hat{\mathbf{b}}), \tag{21}$$

where we define

$$\operatorname{bias}(\hat{\mathbf{b}}) \triangleq \frac{1}{n}\left\|\mathbf{E}\left[\hat{\mathbf{b}}\right]-\mathbf{b}_0\right\|_2^2 \quad \text{and} \quad \operatorname{var}(\hat{\mathbf{b}}) \triangleq \frac{1}{n}\mathbf{E}\left[\left\|\hat{\mathbf{b}}-\mathbf{E}\left[\hat{\mathbf{b}}\right]\right\|_2^2\right]$$

as the bias component and the variance component, respectively.

By Lemma 3, we have $\mathbf{x}^* = \mathbf{V}\mathbf{G}^{-1}\mathbf{U}^T\mathbf{b}$, where $\mathbf{G} = \lambda\mathbf{\Sigma}^{-1}+\mathbf{\Sigma}$. And by Lemma 4, we have $\tilde{\mathbf{x}} = \mathbf{V}\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{b}$, where $\tilde{\mathbf{G}} = \lambda\mathbf{\Sigma}^{-1}+\mathbf{\Sigma}(\mathbf{SV})^T(\mathbf{SV})$. Also notice that $\left\|\mathbf{G}^{-1}\right\|_F^2 = \rho^2$. Using these definition, we first bound the variance component $\operatorname{var}(\tilde{\mathbf{b}})$. We have

$$\begin{aligned}\operatorname{var}(\tilde{\mathbf{b}}) &= \frac{1}{n}\mathbf{E}\left[\left\|\tilde{\mathbf{b}}-\mathbf{E}\left[\hat{\mathbf{b}}\right]\right\|_2^2\right]\\ &= \frac{1}{n}\mathbf{E}\left[\left\|\mathbf{A}\tilde{\mathbf{x}}-\mathbf{E}\left[\mathbf{A}\tilde{\mathbf{x}}\right]\right\|_2^2\right]\\ &= \frac{1}{n}\mathbf{E}\left[\left\|\mathbf{A}\mathbf{V}\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{e}\right\|_2^2\right]\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{n}\mathbf{E}\left[\left\|\mathbf{U}\boldsymbol{\Sigma}\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{e}\right\|_2^2\right] \\
&= \frac{1}{n}\mathbf{E}\left[\operatorname{tr}\left(\mathbf{e}^T\mathbf{U}(\tilde{\mathbf{G}}^{-1})^T\boldsymbol{\Sigma}^2\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{e}\right)\right] \\
&= \frac{1}{n}\mathbf{E}\left[\operatorname{tr}\left(\mathbf{U}(\tilde{\mathbf{G}}^{-1})^T\boldsymbol{\Sigma}^2\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{e}\mathbf{e}^T\right)\right] \\
&= \frac{\sigma^2}{n}\operatorname{tr}\left(\mathbf{U}(\tilde{\mathbf{G}}^{-1})^T\boldsymbol{\Sigma}^2\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\right) \\
&= \frac{\sigma^2}{n}\operatorname{tr}\left((\tilde{\mathbf{G}}^{-1})^T\boldsymbol{\Sigma}^2\tilde{\mathbf{G}}^{-1}\right) \\
&= \frac{\sigma^2}{n}\left\|\boldsymbol{\Sigma}\tilde{\mathbf{G}}^{-1}\right\|_F^2,
\end{aligned}
$$

where we have repeatedly used the cyclic property of trace of matrix product. Similarly, one can show that

$$
\operatorname{var}(\mathbf{b}^*) = \frac{\sigma^2}{n}\left\|\boldsymbol{\Sigma}\mathbf{G}^{-1}\right\|_F^2.
$$

Now we recall the definition $\mathbf{R} = \mathbf{G}^{-1}(\tilde{\mathbf{G}} - \mathbf{G})$ and write $\tilde{\mathbf{G}}^{-1} = (\mathbf{I}+\mathbf{R})^{-1}\mathbf{G}^{-1}$. Then, we have

$$
\begin{aligned}
\left\|\boldsymbol{\Sigma}\tilde{\mathbf{G}}^{-1}\right\|_F &= \left\|\boldsymbol{\Sigma}(\mathbf{I}+\mathbf{R})^{-1}\mathbf{G}^{-1}\right\|_F \\
&= \left\|\boldsymbol{\Sigma}\mathbf{G}^{-1} - \boldsymbol{\Sigma}(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\mathbf{G}^{-1}\right\|_F \qquad (22) \\
&\le \left\|\boldsymbol{\Sigma}\mathbf{G}^{-1}\right\|_F + \left\|\boldsymbol{\Sigma}(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\mathbf{G}^{-1}\right\|_F \qquad (23) \\
&\le \left\|\boldsymbol{\Sigma}\mathbf{G}^{-1}\right\|_F + \left\|\boldsymbol{\Sigma}\right\|_2\left\|(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\right\|_2\left\|\mathbf{G}^{-1}\right\|_F \\
&\le \left\|\boldsymbol{\Sigma}\mathbf{G}^{-1}\right\|_F + \epsilon\rho\left\|\mathbf{A}\right\|_2, \qquad (24)
\end{aligned}
$$

where Eq. (22) follows from Woodbury matrix identity, Eq. (23) follows from triangle inequality and Eq. (24) follows from Lemma 5, the fact that $\mathbf{S}$ is an $(r, \delta, \epsilon/4)$-OSE and the definition that $\rho = \left\|\mathbf{G}^{-1}\right\|_F$.

Now, we can bound $\operatorname{var}(\tilde{\mathbf{b}})$ as follows

$$
\begin{aligned}
\operatorname{var}(\tilde{\mathbf{b}}) &= \frac{\sigma^2}{n}\left\|\boldsymbol{\Sigma}\tilde{\mathbf{G}}^{-1}\right\|_F^2 \\
&\le \frac{\sigma^2}{n}\left(\left\|\boldsymbol{\Sigma}\mathbf{G}^{-1}\right\|_F + \epsilon\rho\left\|\mathbf{A}\right\|_2\right)^2 \\
&\le \frac{\sigma^2}{n}\left(\left\|\boldsymbol{\Sigma}\mathbf{G}^{-1}\right\|_F^2 + 2\epsilon\rho\left\|\mathbf{A}\right\|_2\left\|\boldsymbol{\Sigma}\mathbf{G}^{-1}\right\|_F + \epsilon^2\rho^2\left\|\mathbf{A}\right\|_2^2\right) \\
&\le \frac{\sigma^2}{n}\left(\left\|\boldsymbol{\Sigma}\mathbf{G}^{-1}\right\|_F^2 + 2\epsilon\rho^2\left\|\mathbf{A}\right\|_2^2 + \epsilon^2\rho^2\left\|\mathbf{A}\right\|_2^2\right) \qquad (25) \\
&\le \operatorname{var}(\mathbf{b}^*) + (2\epsilon+\epsilon^2)\frac{\sigma^2}{n}\rho^2\left\|\mathbf{A}\right\|_2^2, \qquad (26)
\end{aligned}
$$

where Eq. (25) follows from the definition $\rho = \left\|\mathbf{G}^{-1}\right\|_F$.

Next, we bound the bias component $\operatorname{bias}(\tilde{\mathbf{b}})$. We can simplify $\operatorname{bias}(\tilde{\mathbf{b}})$ as follows

$$
\begin{aligned}
\operatorname{bias}(\tilde{\mathbf{b}}) &= \frac{1}{n}\left\|\mathbf{E}\left[\tilde{\mathbf{b}}\right] - \mathbf{b}_0\right\|^2 \\
&= \frac{1}{n}\left\|\mathbf{A}\mathbf{V}\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{E}\left[\mathbf{b}\right] - \mathbf{b}_0\right\|_2^2 \\
&= \frac{1}{n}\left\|\mathbf{A}\mathbf{V}\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{A}\mathbf{x}_0 - \mathbf{A}\mathbf{x}_0\right\|_2^2 \\
&= \frac{1}{n}\left\|\mathbf{U}\boldsymbol{\Sigma}\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{A}\mathbf{x}_0 - \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{x}_0\right\|_2^2 \\
&= \frac{1}{n}\left\|\boldsymbol{\Sigma}\tilde{\mathbf{G}}^{-1}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{x}_0 - \boldsymbol{\Sigma}\mathbf{V}^T\mathbf{x}_0\right\|_2
\end{aligned}
$$

$$= \frac{1}{n} \left\|(\boldsymbol{\Sigma}\tilde{\mathbf{G}}^{-1} - \mathbf{I})\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{x}_0\right\|_2^2,$$

where we have dropped $\mathbf{U}$ which does not change l_2 norms. Using the same method, $\mathrm{bias}(\mathbf{b}^*)$ can be simplified as

$$\mathrm{bias}(\mathbf{b}^*) = \frac{1}{n} \left\|(\boldsymbol{\Sigma}\mathbf{G}^{-1} - \mathbf{I})\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{x}_0\right\|_2^2.$$

It is easy to see that $\sqrt{n\,\mathrm{bias}(\tilde{\mathbf{b}})} = \left\|(\boldsymbol{\Sigma}\tilde{\mathbf{G}}^{-1} - \mathbf{I})\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{x}_0\right\|_2$ and we can bound it as follows

$$\begin{aligned}
\sqrt{n\,\mathrm{bias}(\tilde{\mathbf{b}})} &= \left\|(\boldsymbol{\Sigma}\tilde{\mathbf{G}}^{-1} - \mathbf{I})\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{x}_0\right\|_2 \\
&= \left\|(\boldsymbol{\Sigma}(\mathbf{I}+\mathbf{R})^{-1}\mathbf{G}^{-1} - \mathbf{I})\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{x}_0\right\|_2 \\
&= \left\|(\boldsymbol{\Sigma}\mathbf{G}^{-1} - \mathbf{I} - \boldsymbol{\Sigma}(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\mathbf{G}^{-1})\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{x}_0\right\|_2 && (27)\\
&\le \left\|(\boldsymbol{\Sigma}\mathbf{G}^{-1} - \mathbf{I})\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{x}_0\right\|_2 + \left\|\boldsymbol{\Sigma}(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\mathbf{G}^{-1}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{x}_0\right\|_2 && (28)\\
&\le \sqrt{n\,\mathrm{bias}(\mathbf{b}^*)} + \|\boldsymbol{\Sigma}\|_2 \left\|(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\right\|_2 \left\|\mathbf{G}^{-1}\boldsymbol{\Sigma}\right\|_2 \left\|\mathbf{V}^T\mathbf{x}_0\right\|_2 \\
&\le \sqrt{n\,\mathrm{bias}(\mathbf{b}^*)} + \|\mathbf{A}\|_2 \cdot \epsilon \cdot \max_i \frac{\sigma_i^2}{\lambda+\sigma_i^2} \|\mathbf{x}_0\|_2 && (29)\\
&\le \sqrt{n\,\mathrm{bias}(\mathbf{b}^*)} + \epsilon \|\mathbf{A}\|_2 \|\mathbf{x}_0\|_2, && (30)
\end{aligned}$$

where Eq. (27) follows from matrix inversion lemma, Eq. (28) is the triangle inequality and Eq. (29) follows from Lemma 5 and that $\mathbf{S}$ is an $(r, \delta, \epsilon/4)$-OSE. Now, dividing both sides of the above inequality Eq. (30) by $\sqrt{n}$, we have that

$$\sqrt{\mathrm{bias}(\tilde{\mathbf{b}})} \le \sqrt{\mathrm{bias}(\mathbf{b}^*)} + \frac{\epsilon}{\sqrt{n}} \|\mathbf{A}\|_2 \|\mathbf{x}_0\|_2.$$

Finally, we obtain the following bound on $\mathrm{bias}(\tilde{\mathbf{b}})$

$$\begin{aligned}
\mathrm{bias}(\tilde{\mathbf{b}}) &\le \mathrm{bias}(\mathbf{b}^*) + 2\sqrt{\mathrm{bias}(\mathbf{b}^*)} \cdot \frac{\epsilon}{\sqrt{n}} \|\mathbf{A}\|_2 \|\mathbf{x}_0\|_2 + \frac{\epsilon^2}{n} \|\mathbf{A}\|_2^2 \|\mathbf{x}_0\|_2^2 \\
&\le \mathrm{bias}(\mathbf{b}^*) + \frac{2}{\sqrt{n}} \left\|(\boldsymbol{\Sigma}\mathbf{G}^{-1} - \mathbf{I})\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{x}_0\right\|_2 \cdot \frac{\epsilon}{\sqrt{n}} \|\mathbf{A}\|_2 \|\mathbf{x}_0\|_2 + \frac{\epsilon^2}{n} \|\mathbf{A}\|_2^2 \|\mathbf{x}_0\|_2^2 \\
&\le \mathrm{bias}(\mathbf{b}^*) + 2\left(\frac{1}{\sqrt{n}} \left\|\boldsymbol{\Sigma}\mathbf{G}^{-1} - \mathbf{I}\right\|_2 \|\boldsymbol{\Sigma}\|_2 \left\|\mathbf{V}^T\mathbf{x}_0\right\|_2\right)\left(\frac{\epsilon}{\sqrt{n}} \|\mathbf{A}\|_2 \|\mathbf{x}_0\|_2\right) + \frac{\epsilon^2}{n} \|\mathbf{A}\|_2^2 \|\mathbf{x}_0\|_2^2 \\
&\le \mathrm{bias}(\mathbf{b}^*) + \frac{2\epsilon}{n} \|\mathbf{A}\|_2^2 \|\mathbf{x}_0\|_2^2 + \frac{\epsilon^2}{n} \|\mathbf{A}\|_2^2 \|\mathbf{x}_0\|_2^2 && (31)\\
&= \mathrm{bias}(\mathbf{b}^*) + \frac{(2\epsilon+\epsilon^2)}{n} \|\mathbf{A}\|_2^2 \|\mathbf{x}_0\|_2^2, && (32)
\end{aligned}$$

where Eq. (31) holds since $\|\boldsymbol{\Sigma}\|_2 = \|\mathbf{A}\|_2$, $\left\|\mathbf{V}^T\mathbf{x}_0\right\|_2 = \|\mathbf{x}_0\|_2$ and $\left\|\boldsymbol{\Sigma}\mathbf{G}^{-1} - \mathbf{I}\right\|_2 = \max_i \frac{\lambda}{\lambda+\sigma_i^2} \le 1$.

The theorem follows immediately from Eq. (21), Eq. (26), Eq. (32) and the fact that $\epsilon < 1$. □

C Proof of Theorem 7

We first prove a generalization of Lemma 1 as follows.

Lemma 7. *Given $\mathbf{A} \in \mathbb{R}^{n\times p}$ of rank r, $\mathbf{B} \in \mathbb{R}^{n\times m}$ and $\lambda > 0$. Suppose that $\mathbf{S} \in \mathbb{R}^{t\times p}$ is an $(r, \delta, \epsilon/4)$-OSE for $\epsilon \in (0, 1)$ and $\delta \in (0, 1)$. Then, with probability $1 - \delta$, we have*

$$\left\|\tilde{\mathbf{X}} - \mathbf{X}^*\right\|_F \le \epsilon \|\mathbf{X}^*\|_F,$$

where $\tilde{\mathbf{X}}$ is given by Eq. (16) and $\mathbf{X}^*$ is the optimal solution to multiple response ridge regression Eq. (14).

Proof. Let the thin SVD of $\mathbf{A}$ be $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Fix an arbitrary $i \in [m]$, consider the column vectors $\tilde{\mathbf{X}}^{(i)}$ and $\mathbf{X}^{*(i)}$. By definition, we can see that $\mathbf{X}^{*(i)} = \mathbf{A}^T(\lambda\mathbf{I}_n + \mathbf{A}\mathbf{A}^T)^{-1}\mathbf{B}^{(i)}$ and

$$\tilde{\mathbf{X}}^{(i)} = \mathbf{A}^T(\mathbf{A}\mathbf{S}^T)^{\dagger^T}(\lambda(\mathbf{A}\mathbf{S}^T)^{\dagger^T} + \mathbf{A}\mathbf{S}^T)^{\dagger}\mathbf{B}^{(i)}.$$

Define $\tilde{\mathbf{G}} = \lambda\mathbf{\Sigma}^{-1} + \mathbf{\Sigma}(\mathbf{S}\mathbf{V})^T(\mathbf{S}\mathbf{V})$ and $\mathbf{G} = \lambda\mathbf{\Sigma}^{-1} + \mathbf{\Sigma}$. Now, we can regard $\mathbf{B}^{(i)}$ as the target vector and then apply Lemma 3 on $\mathbf{X}^{*(i)}$ and Lemma 4 on $\tilde{\mathbf{X}}^{(i)}$, respectively. This shows that $\mathbf{X}^{*(i)} = \mathbf{V}\mathbf{G}^{-1}\mathbf{U}^T\mathbf{B}^{(i)}$ and $\tilde{\mathbf{X}}^{(i)} = \mathbf{V}\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{B}^{(i)}$. Hence, combining all columns $i \in [m]$, we have that $\mathbf{X}^* = \mathbf{V}\mathbf{G}^{-1}\mathbf{U}^T\mathbf{B}$ and $\tilde{\mathbf{X}} = \mathbf{V}\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{B}$.

Similar to the proof of Lemma 1, we recall the definition $\mathbf{R} = \mathbf{G}^{-1}(\tilde{\mathbf{G}} - \mathbf{G})$ and write $\tilde{\mathbf{G}}^{-1} = (\mathbf{I} + \mathbf{R})^{-1}\mathbf{G}^{-1}$. Let $\epsilon' = \epsilon/4$. Then, we have

$$\begin{aligned}
\left\|\tilde{\mathbf{X}} - \mathbf{X}^*\right\|_F &= \left\|\mathbf{V}(\tilde{\mathbf{G}}^{-1} - \mathbf{G}^{-1})\mathbf{U}^T\mathbf{B}\right\|_F \\
&= \left\|(\tilde{\mathbf{G}}^{-1} - \mathbf{G}^{-1})\mathbf{U}^T\mathbf{B}\right\|_F \\
&= \left\|\left((\mathbf{I} + \mathbf{R})^{-1} - \mathbf{I}\right)\mathbf{G}^{-1}\mathbf{U}^T\mathbf{B}\right\|_F \\
&= \left\|(\mathbf{I} + \mathbf{R})^{-1}\mathbf{R}\mathbf{G}^{-1}\mathbf{U}^T\mathbf{B}\right\|_F \qquad (33) \\
&\le \left\|(\mathbf{I} + \mathbf{R})^{-1}\mathbf{R}\right\|_2 \left\|\mathbf{G}^{-1}\mathbf{U}^T\mathbf{B}\right\|_F \\
&= \left\|(\mathbf{I} + \mathbf{R})^{-1}\mathbf{R}\right\|_2 \left\|\mathbf{X}^*\right\|_F \\
&\le \frac{2\epsilon' + \epsilon'^2}{1 - (2\epsilon' + \epsilon'^2)} \left\|\mathbf{X}^*\right\|_F \qquad (34) \\
&\le 4\epsilon' \left\|\mathbf{X}^*\right\|_F \\
&= \epsilon \left\|\mathbf{X}^*\right\|_F . \qquad (35)
\end{aligned}$$

where Eq. (33) follows from matrix inversion lemma and Eq. (34) follows from Lemma 5 and the fact that $\mathbf{S}$ is an $(r, \delta, \epsilon/4)$-OSE. □

Then, Theorem 7 is an immediate consequence of Lemma 7.

Proof of Theorem 7. The bound on $\left\|\tilde{\mathbf{X}} - \mathbf{X}\right\|_F$ follows immediately from Lemma 7 and Theorem 3 which shows that $\mathbf{S} = \mathbf{\Phi}_{\text{srht}}\mathbf{\Phi}_{\text{sparse}}$ is an $(r, \delta, \epsilon/4)$-OSE. And the running time analysis is similar to the one given in Section 3. □

D Structured Ridge Regression: Algorithm Details and Analysis

In this part, we supplement the details of the relative-error approximation algorithm for structured ridge regression.

First, we compute the sketched matrix $\varphi(\mathbf{A})\mathbf{S}^T$, where $\mathbf{S} = \mathbf{\Phi}_{\text{srht}}\mathbf{\Phi}_{\text{sparse}}$. Next, we compute the approximate solution $\tilde{\mathbf{x}}$ as follows

$$\tilde{\mathbf{x}} = \varphi(\mathbf{A})^T(\varphi(\mathbf{A})\mathbf{S}^T)^{\dagger^T}(\lambda(\varphi(\mathbf{A})\mathbf{S}^T)^{\dagger^T} + \varphi(\mathbf{A})\mathbf{S}^T)^{\dagger}\mathbf{b}. \qquad (36)$$

The following theorem shows that the $\tilde{\mathbf{x}}$ obtained by Eq. (36) is a relative-error approximation of $\mathbf{x}^*$.

Theorem 8. *Given $\mathbf{A} \in \mathbb{R}^{n\times p}$, $\mathbf{b} \in \mathbb{R}^n$, $\lambda > 0$, $q \in \mathbb{N}^+$, parameter $\epsilon \in (0, 1)$ and $\delta \in (0, 1)$. Select integers t', t such that $t' \ge 2\delta^{-1}(n^2 + n)/(\epsilon/6 - \epsilon^2/144)^2$ and $t \ge 72\epsilon^{-1}[\sqrt{n} + \sqrt{8\log(6p/\delta)}]^2\log(6n/\delta)$. Let $\mathbf{S} = \mathbf{\Phi}_{srht}\mathbf{\Phi}_{sparse}$, where $\mathbf{\Phi}_{sparse} \in \mathbb{R}^{t'\times pq}$ is a sparse embedding matrix and $\mathbf{\Phi}_{srht} \in \mathbb{R}^{t\times t'}$ is an SRHT matrix. Then, with probability at least $1 - \delta$, we have*

$$\left\|\tilde{\mathbf{x}} - \mathbf{x}^*\right\|_2 \le \epsilon\left\|\mathbf{x}^*\right\|_2 ,$$

where $\tilde{\mathbf{x}}$ is given by Eq. (36) and $\mathbf{x}^*$ is the optimal solution to structured ridge regression Eq. (17). In addition, the total time complexity of computing $\varphi(\mathbf{A})\mathbf{S}^T$ and $\tilde{\mathbf{x}}$ is $O(\text{nnz}(\mathbf{A})\log^2(q) + n^3q\log^2(q)/\epsilon^2 + n^3\log(\frac{n}{\epsilon})/\epsilon^2)$.

In order to prove Theorem 8, we only need to calculate the running time of Eq. (36), since the relative error approximation guarantee directly comes from Theorem 4. To analyze the running time, we use the following result of Avron et al. [3].

Lemma 8. [3, Lemma 9, 10, 11] *Let $\mathbf{A} \in \mathbb{R}^{n\times p}$, $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{z} \in \mathbb{R}^{pq}$. Let $\varphi_q : \mathbb{R}^p \to \mathbb{R}^{pq}$ be the kernel expansion function such that $\varphi_q(\mathbf{a}) = \{a_i^{j-1}\}_{(i,j)\in[p]\times[q]}$ for all $\mathbf{a} \in \mathbb{R}^p$. Let $\mathbf{\Phi}_{sparse} \in \mathbb{R}^{t'\times pq}$ be a sparse embedding matrix.*

Then, there exists a fast matrix-vector multiplication algorithm such that $\mathbf{y}^T\varphi_q(\mathbf{A})$ *and* $\varphi_q(\mathbf{A})\mathbf{z}$ *can be computed in* $O(\mathrm{nnz}(\mathbf{A})\log^2(q) + nq\log^2(q))$ *time. And therefore the product* $\varphi_q(\mathbf{A})\mathbf{\Phi}^T_{sparse}$ *can be computed in* $O(\mathrm{nnz}(\mathbf{A})\log^2(q) + nqt'\log^2(q))$ *time.*

Proof of Theorem 8. From Theorem 3 we knows that $\mathbf{S} = \mathbf{\Phi}_{\text{srht}}\mathbf{\Phi}_{\text{sparse}}$ is an $(n, \delta, \epsilon/4)$ OSE. The bound on $\|\tilde{\mathbf{x}} - \mathbf{x}\|_2$ follows immediately from Lemma 1.

From Lemma 8 we know the time cost to compute $\varphi_q(\mathbf{A})\mathbf{\Phi}^T_{\text{sparse}}$ is $O((\mathrm{nnz}(\mathbf{A}) + qn^3/\epsilon^2)\log^2(q))$. Therefore, the total time complexity of computing $\varphi_q(\mathbf{A})\mathbf{S}^T$ and $\tilde{\mathbf{x}}$ is $O(\mathrm{nnz}(\mathbf{A})\log^2(q) + qn^3/\epsilon^2\log^2(q) + n^3/\epsilon^2\log(\frac{n}{\epsilon}))$. □

E Argument and Counterexample for the Proof of [22, Theorem 1]

In this section, we show a counterexample of a claim used in [22, Theorem 1]. For clarity, we will use the notations in [22].

The risk inflation bound of [22] is based on the claim that the following function

$$B(\mathbf{M}) = n\lambda^2\mathbf{z}^T(\mathbf{M} + n\lambda\mathbf{I}_n)^{-2}z$$

is non-increasing in $\mathbf{M}$ for any positive semi-definite matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ and $\mathbf{z} \in \mathbb{R}^n$. It is easy to see that this claim is equivalent to the claim that for any positive definite matrices $\mathbf{M}_1, \mathbf{M}_2 \in \mathbb{R}^{n\times n}$, if $\mathbf{M}_1 \succeq \mathbf{M}_2 \succ \mathbf{0}_n$, then $\mathbf{M}_1^{-2} \preceq \mathbf{M}_2^{-2}$.

However, this claim is not true and we can construct following counterexample.

Let $\mathbf{M}_1$ and $\mathbf{M}_2$ be the following 2×2 matrices.

$$\mathbf{M}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix}, \quad \mathbf{M}_2 = \begin{bmatrix} 2/3 & -2 \\ -2 & 12 \end{bmatrix}.$$

It is easy to verify that $\mathbf{M}_1 \succeq \mathbf{M}_2 \succ \mathbf{0}_2$, but $\mathbf{M}_1^{-2} \npreceq \mathbf{M}_2^{-2}$. This issue also propagates to other parts of their proof. Hence their bound [22, Theorem 1] may be flawed.