Randomized Iterative Algorithms for Fisher Discriminant Analysis

Agniva Chowdhury  
Department of Statistics  
Purdue University  
West Lafayette, IN  
chowdhu5@purdue.edu

Jiasen Yang  
Department of Statistics  
Purdue University  
West Lafayette, IN  
jiaseny@purdue.edu

Petros Drineas  
Department of Computer Science  
Purdue University  
West Lafayette, IN  
pdrineas@purdue.edu

Abstract

Fisher discriminant analysis (FDA) is a widely used method for classification and dimensionality reduction. When the number of predictor variables greatly exceeds the number of observations, one of the alternatives for conventional FDA is regularized Fisher discriminant analysis (RFDA). In this paper, we present a simple, iterative, sketching-based algorithm for RFDA that comes with provable accuracy guarantees when the number of predictor variables greatly exceeds the number of observations, one of the alternatives for conventional approaches and our empirical evaluations support our theoretical analyses.

1 INTRODUCTION

In multivariate statistics and machine learning, Fisher’s linear discriminant analysis (FDA) is a widely used method for classification and dimensionality reduction. The main idea is to project the data onto a lower dimensional space such that the separability of points between the different classes is maximized while the separability of points within each class is minimized.

Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ be the centered data matrix whose rows represent $n$ points in $\mathbb{R}^d$. We assume that $\mathbf{A}$ is centered around $\mathbf{m} \in \mathbb{R}^d$, with $\mathbf{m}$ being the grand-mean of the original raw (non-centered) data-points.¹ Suppose there are $c$ disjoint classes with $n_j$ observations belonging to the $j$-th class and $\sum_{j=1}^c n_j = n$. Further, let $\mathbf{m}_j \in \mathbb{R}^d$ denote the mean vector of the raw (non-centered) data-points corresponding to the $j$-th class, $j = 1, 2, \ldots, c$.

Define the total scatter matrix

$$
\Sigma_t \triangleq \sum_{i=1}^n (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T \in \mathbb{R}^{d \times d},
$$

where $\mathbf{x}_i$ is the $i$-th raw data-point. Similarly, define the between-class scatter matrix

$$
\Sigma_b \triangleq \sum_{j=1}^c n_j (\mathbf{m}_j - \mathbf{m})(\mathbf{m}_j - \mathbf{m})^T \in \mathbb{R}^{d \times d}.
$$

Under these notations, conventional FDA solves the generalized eigen-problem

$$
\Sigma_b \mathbf{x}_i = \lambda_i \Sigma_t \mathbf{x}_i, \quad i = 1, 2, \ldots, q,
$$

where $\mathbf{x}_i$ is called the $i$-th discriminant direction, with $q \leq \min\{d, c-1\}$ and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_q > 0$. We can further express this problem in matrix form as

$$
\Sigma_b \mathbf{X} = \Sigma_t \mathbf{X} \mathbf{A},
$$

(1)

where $\mathbf{X} \triangleq [\mathbf{x}_1 \mathbf{x}_2 \cdots \mathbf{x}_q] \in \mathbb{R}^{d \times q}$ and $\mathbf{A} \triangleq \text{diag}\{\lambda_1, \ldots, \lambda_q\}$. An elegant linear algebraic formulation of eqn. (1) was presented in [36]:

$$
(\mathbf{A}^T \Omega \mathbf{A}) \mathbf{X} = (\mathbf{A}^T \mathbf{A}) \mathbf{X} \Lambda,
$$

(2)

where $\Sigma_t = \mathbf{A}^T \mathbf{A}$ and $\Sigma_b = \mathbf{A}^T \Omega \mathbf{A}$. Here, $\Omega \in \mathbb{R}^{n \times c}$ denotes the rescaled class membership matrix, with $\Omega_{ij} = 1/\sqrt{n_j}$ if the $i$-th row of $\mathbf{A}$ (i.e., the $i$-th data point) is a member of the $j$-th class; otherwise $\Omega_{ij} = 0$.

¹If the original data were represented by the matrix $\hat{\mathbf{A}} \in \mathbb{R}^{n \times d}$, then $\mathbf{m}$ is the row-wise mean of $\hat{\mathbf{A}}$ and $\mathbf{m} = \hat{\mathbf{A}} - 1_n \mathbf{1}_n^T$, where $1_n$ is the all-ones vector. As a result of mean-centering, $\text{rank}(\mathbf{A}) \leq \min\{n-1, d-1\}$. 

We present a novel iterative, sketching-based algorithm where (The last equality can be verified using the SVD of $A$ which replaces $A$). This implies that instead of using the actual solution.

In RFDA, $(A^T A)^{-1}$ is replaced by $(A^T A + \lambda I_d)^{-1}$, where $\lambda > 0$ is a regularization parameter. In this case, eqn. (2) becomes

$$G \Omega^T A X = X \Lambda,$$

where

$$G = (A^T A + \lambda I_d)^{-1} A^T \Omega = A^T (A A^T + \lambda I_n)^{-1} \Omega.$$ (The last equality can be verified using the SVD of $A$.) Note that $A^T A + \lambda I_d$ is always invertible for $\lambda > 0$. We define the effective degrees of freedom of RFDA as

$$d_\lambda = \sum_{i=1}^\rho \frac{\sigma_i^2}{\sigma_i^2 + \lambda} \leq \rho .$$

Here, $\rho$ is the rank of the matrix $A$ and we note that $d_\lambda$ depends on both the value of the regularization parameter $\lambda$ and the non-zero singular values $\sigma_i^2, i = 1, 2, \ldots, \rho$.

Solving the RFDA problem of eqn. (3). Notice that the solution $(X, \Lambda)$ to eqn. (3) may not be unique. Indeed, if $X$ is a solution to eqn. (3), then for any non-singular diagonal matrix $D \in \mathbb{R}^{d \times d}$, $XD$ is also a solution [36] of an eigenvalue decomposition (EVD)-based algorithm (see Algorithm 2 in Appendix B) which not only returns $X$ as a solution to eqn. (3) but also guarantees that for any two data points $w_1, w_2 \in \mathbb{R}^d$, $X$ satisfies $\| (w_1 - w_2)^T X \|_2 = \| (w_1 - w_2)^T G \|_2$ (see Theorem 8). This implies that instead of using the actual solution $X$, if we project the points using $G$, the distances between the projected points would also be preserved. Thus, for any distance-based classification method (e.g., k-nearest-neighbors), both $X$ and $G$ would result in the same predictions. Therefore, when solving eqn. (3) it is reasonable to shift our interest from $X$ to $G$. However, due to the high dimensionality $d$ of the input data, exact computation of $G$ is expensive, taking time $O(n^2 d + n^3 + ndc)$.

### 1.1 OUR CONTRIBUTIONS

We present a novel iterative, sketching-based algorithm for the RFDA problem that guarantees highly accurate solutions when compared to conventional approaches. Our analysis builds upon simple structural conditions that boil down to randomized matrix multiplication, a fundamental and well-understood primitive of randomized linear algebra. Our main algorithm (see Algorithm 1) is analyzed in light of the following structural constraint, which constructs a sketching matrix $S \in \mathbb{R}^{s \times d}$ (for an appropriate choice of the sketching dimension $s \ll d$), such that

$$\| \Sigma_\lambda V^T S S^T V \Sigma_\lambda - \Sigma_\lambda^2 \|_2 \leq \frac{\varepsilon}{2}.$$

Here, $V \in \mathbb{R}^{d \times \rho}$ contains the right singular vectors of $A$ and $\Sigma_\lambda \in \mathbb{R}^{\rho \times \rho}$ is a diagonal matrix with

$$\Sigma(i,i) = \frac{\sigma_i^2}{\sigma_i^2 + \lambda}, \quad i = 1, \ldots, \rho .$$

Notice that $\| \Sigma_\lambda^2 \|_F^2 = d_\lambda$, which is defined to be the effective degrees of freedom of the RFDA problem (see eqn. (4)). Eqn. (5) can be satisfied by sampling with respect to the ridge leverage scores of $[2, 8]$ (cf. Section 1.3) or by oblivious sketching matrix constructions (e.g., count-sketch [7] or sub-sampled randomized Hadamard transform (SRHT) [1, 14, 30]) for $S$ with sketch-size $s$ depending on $d_\lambda$ (see Appendix F for details). Recall that $d_\lambda$ is upper bounded by $\rho$ but could be significantly smaller depending on the distribution of the singular values and the choice of $\lambda$. Indeed, it follows that by sampling-and-rescaling $O(d_\lambda \ln d_\lambda)$ predictor variables from the matrix $A$ (using either exact or approximate ridge leverage scores [2, 8]), we can satisfy the constraint of eqn. (5), and Algorithm 1 would yield an estimator $G$ satisfying

$$\| (w - m)^T (G - G) \|_2 \leq \frac{\varepsilon}{\sqrt{\lambda}} \|VV^T(w - m)\|_2 .$$

Here, $w \in \mathbb{R}^d$ is any test data point and $VV^T(w - m)$ is the part of $w - m$ that lies within the range of $A^T$ (see footnote 1 for the definition of $m$). We note that the dependency of the error on $\varepsilon$ drops exponentially fast as the number of iterations $t$ increases. See Section 2.2 for constructions of $S$ and Section 1.2 for a comparison of this bound with prior work.

Additionally, we complement the bound of eqn. (7) with a second bound subject to a different structural condition:

$$\| V^T S S^T V - I_\rho \|_2 \leq \frac{\varepsilon}{2} .$$

Indeed, assuming that the rank of $A$ is much smaller than $\min\{n, d\}$, one can use the (exact or approximate) column leverage scores [22, 21] of the matrix $A$ (cf. Section 1.3) to satisfy the aforementioned constraint by sampling $O(\rho \ln \rho)$ columns, in which case $S$ is a sampling-and-rescaling matrix. Perhaps more interestingly, a variety of oblivious sketching matrix constructions for $S$ can also be used to satisfy eqn. (8) (see Section 2.2 for
specific constructions of $S$). In either case, under this structural condition, the output of Algorithm 1 satisfies
\[ \|(w - m)^T(\hat{G} - G)\|_2 \leq \frac{\varepsilon}{2\sqrt{\lambda}} \|VV^T(w - m)\|_2. \]
(9)

The above guarantee is essentially identical to that of eqn. (7), with the approximation error decaying exponentially fast as the number of iterations $t$ increases. However, eqn. (8) exhibits a worse dependency on the sketch size $s$. Indeed, eqn. (8) can be satisfied by sampling-and-rescaling $O(\rho \ln \rho)$ predictor variables from the matrix $A$, which could be much larger than the sketch size needed when sampling with respect to the ridge leverage scores.

To the best of our knowledge, our bounds are the first attempt to provide general structural results that guarantee provable, high-quality solutions for the RFDA problem.

To summarize, our first structural result (Theorem 1) can be satisfied by sampling with respect to ridge leverage scores or by the use of oblivious sketching matrices whose size depends on the effective degrees of freedom, yielding a highly accurate guarantee in terms of “distance distortion” caused by iterative sketching. While ridge leverage scores have been used in a number of applications including matrix approximation, cost-preserving projections, and $k$-means clustering [8], their performance in the context of RFDA has not been analyzed in prior work. Our second structural result (Theorem 2) complements the analysis of Theorem 1 subject to a second structural condition (eqn. (8)) which can be satisfied by sampling with respect to standard leverage scores using a sketch size that depends on the rank of the centered data matrix.

1.2 PRIOR WORK

The work most closely related to ours is [32], where the authors proposed a fast random projection–based algorithm to accelerate RFDA. Their theoretical analysis showed that random projections (and in particular the count-min sketch) preserve the generalization ability of FDA on the original training data. However, for the $d \gg n$ case, the error bound in their work (Theorem 3 of [32]) depends on the condition number of the centered data matrix $A$. More precisely, they proved that their method computes a matrix $\hat{G}$ in time $O(\min(A)) + O(n^2 s + n^3 + n de)$, \(^3\) which, for any test data point $w \in \mathbb{R}^d$, satisfies
\[ \|(w - m)^T(\hat{G} - G)\|_2 \leq \frac{\kappa \varepsilon}{1 - \varepsilon} \|VV^T(w - m)\|_2, \]
with high probability for any $\varepsilon \in (0, 1]$ (here, $\kappa$ is the condition number of $A$). Thus, their random projection–based approach well-approximates the original RFDA problem only when $A$ is well-conditioned ($i.e., \kappa$ small).

In addition to the running time of their approach grows proportionally to $O(1/\varepsilon^2)$, whereas our algorithm runs in $O(\log(1/\varepsilon))$ time (cf. Section 2.2). Lastly, our main result depends only on the effective degrees of freedom $d\lambda$ (cf. Theorem 1), which can be much smaller than $\rho$.

Our work was inspired by [36], where the authors presented a flexible and efficient implementation of RFDA through an EVD-based algorithm. In addition, [36] uncovered a general relationship between RFDA and ridge regression that explains how matrix $G$ has similar properties with the solution matrix $X$ in terms of distance-based classification methods. We also note that using their linear algebraic formulation and the proposed EVD-based framework, [32] presented a fast implementation of FDA. Another line of work that motivated our approach was the framework of leverage score sampling and the relatively recent introduction of ridge leverage scores [2, 8]. Indeed, our Theorems 1 and 2 present structural results that can be satisfied (with high probability) by sampling columns of $A$ with probabilities proportional to (exact or approximate) ridge leverage scores and leverage scores, respectively (see Section 2.2). To the best of our knowledge, these are the first results providing a strong accuracy guarantee for RFDA problems when ridge leverage scores are used to sample predictor variables.

Under a different context, a recent paper [6] presented an iterative algorithm for solving ridge regression problems with $d \gg n$ in a sketching–based framework. There, the authors proved that the output of their proposed algorithm closely approximates the true solution of the ridge regression problem if the columns of the data matrix are sampled with probabilities proportional to the column ridge leverage scores. While the results in [6] require assumptions on $\lambda$ and the singular values of $A$, a key advantage of the present work is that our main result (Theorem 1) is valid for any $\lambda > 0$. From the sketching perspective, we also emphasize that the distinction between regularized regression problems and FDA is substantial.

Among other relevant works, [27] proposed an iterative algorithm for ridge regression that unifies (and accelerates) the so-called iterative Hessian sketch (IHS) [23] and iterative dual random projection (IDRP) [35] together to reduce the number of observations and dimensionality simultaneously. However, it is not straightforward to extend the idea of [27] in an FDA-based classification framework. In another paper [25], the authors addressed the scalability of FDA by developing a random projection–based FDA algorithm and presented a theoretical analysis of the approximation error involved. However, their framework applies exclusively to the two-stage FDA problem [4, 34], where the issue of singularity is addressed before

\(^3\)Here, $s = O(\rho^2/\delta^2)$, where $\delta$ is the failure probability.
We shall also make use of the null-space based FDA [5] for \( d \gg n \) using random matrices. Nevertheless, their approach is quite different from ours and does not come with provable guarantees. In addition, [15] provided a tight bound on classification error when FDA is applied in a random projection–based reduced feature-space. However, their approach utilizes the within-class scatter matrix \( \Sigma_i - \Sigma_b \), which becomes costly to compute and potentially ill-conditioned when \( d \) is large, resulting in unreliable predictions. Finally, [33] proposed an iterative approach to address the singularity of \( \Lambda^T \Lambda \), where the underlying data representation model is different from conventional FDA. Their approach does not yield a closed form solution for the discriminant directions.

1.3 NOTATION

We use \( a, b, \ldots \) to denote vectors and \( A, B, \ldots \) to denote matrices. For a matrix \( A \), \( A_{i:t} \) (\( A_{i:t} \)) denotes the \( i \)-th column (row) of \( A \) as a column (row) vector. For a vector \( a \), \( \|a\|_2 \) denotes its Euclidean norm; for a matrix \( A \), \( \|A\|_F \) denotes its Frobenius norm. We refer the reader to [18] for properties of norms that will be quite useful in our work.

For a matrix \( A \in \mathbb{R}^{n \times d} \) with \( d \geq n \) of rank \( \rho \), its (thin) Singular Value Decomposition (SVD) is the product \( U \Sigma V^T \), with \( U \in \mathbb{R}^{n \times \rho} \) (the matrix of the left singular vectors), \( V \in \mathbb{R}^{d \times \rho} \) (the matrix of the right singular vectors), and \( \Sigma \in \mathbb{R}^{\rho \times \rho} \) a diagonal matrix whose diagonal entries are the non-zero singular values of \( A \) arranged in non-increasing order. Computation of the SVD takes, in this setting, \( \mathcal{O}(n^2d) \) time. We will often use \( \sigma_i \) to denote the singular values of a matrix implied by context.

We shall also make use of the full SVD representation \( A = U_f \Sigma_f V_f^T \), where \( U_f = (U_\| U_\perp) \in \mathbb{R}^{n \times n} \), \( V_f = (V \perp V_\perp) \in \mathbb{R}^{d \times d} \), and \( \Sigma_f = \begin{pmatrix} \Sigma_{\perp \perp} & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{n \times d} \).

Here, \( U_\perp \in \mathbb{R}^{n \times (n - \rho)} \) and \( V_\perp \in \mathbb{R}^{d \times (d - \rho)} \). Finally, the column leverage scores and ridge leverage scores of \( A \) are respectively given by \( \|V_{\perp i}\|_2^2 \) and \( \|(V \Sigma_\perp)_{i*}\|_2^2 \) for \( i = 1, 2, \ldots, d \). (Recall the definition of \( \Sigma_\lambda \) in eqn. (6).) Additional notation will be introduced as needed.

2 ITERATIVE, SKETCHED FISHER DISCRIMINANT ANALYSIS

2.1 AN ITERATIVE, SKETCHING-BASED ALGORITHM

Our main algorithm (Algorithm 1) solves a sketched RFDA problem in each iteration updating the (rescaled) class membership matrix to account for the information already captured in prior iterations. More precisely, our algorithm iteratively computes a sequence of matrices \( \tilde{G}^{(j)} \in \mathbb{R}^{d \times c} \) for \( j = 1, \ldots, t \) and returns the estimator \( \hat{G} = \sum_{j=1}^t \tilde{G}^{(j)} \) to the original matrix \( G \) of eqn. (3). Our main quality-of-approximation results (Theorems 1 and 2) argue that returning the sum of those intermediate matrices results in a highly accurate approximation to the direct RFDA solution.

Algorithm 1 Iterative RFDA Sketch

**Input:** \( A \in \mathbb{R}^{n \times d} \), \( \Omega \in \mathbb{R}^{n \times c} \), \( \lambda > 0 \); number of iterations \( t > 0 \); sketching matrix \( S \in \mathbb{R}^{d \times s} \).

**Initialize:** \( L^{(0)} \leftarrow \Omega \), \( G^{(0)} \leftarrow 0_{d \times c} \), \( Y^{(0)} \leftarrow 0_{n \times c} \).

**for** \( j = 1 \) to \( t \) **do**

\[ L^{(j)} \leftarrow L^{(j-1)} - \lambda Y^{(j-1)} - A \tilde{G}^{(j-1)}; \]

\[ Y^{(j)} \leftarrow (A^T A + \lambda I_n)^{-1} L^{(j)}; \]

\[ G^{(j)} \leftarrow A^T Y^{(j)}; \]

**end for**

**Output:** \( \hat{G} = \sum_{j=1}^t \tilde{G}^{(j)}; \)

Theorem 1 presents our approximation guarantees under the assumption that the sketching matrix \( S \) satisfies the constraint of eqn. (5).

**Theorem 1.** Let \( A \in \mathbb{R}^{n \times d} \) and \( G \in \mathbb{R}^{d \times c} \) be as defined in Section 1. Assume that for some constant \( 0 < \varepsilon < 1 \) the sketching matrix \( S \in \mathbb{R}^{d \times s} \) satisfies eqn. (5). Then, for any test data point \( w \in \mathbb{R}^d \), the estimator \( \hat{G} \) returned by Algorithm 1 satisfies

\[ \| (w - m)^T (\hat{G} - G) \|_2 \leq \frac{\varepsilon t}{\sqrt{\lambda}} \| V V^T (w - m) \|_2. \]

Recall that \( V V^T (w - m) \) is the projection of the vector \( w - m \) onto the row space of \( A \).

Similarly, Theorem 2 presents our accuracy guarantees under the assumption that the sketching matrix \( S \) satisfies the constraint of eqn. (8).

**Theorem 2.** Let \( A \in \mathbb{R}^{n \times d} \) and \( G \in \mathbb{R}^{d \times c} \) be as defined in Section 1. Assume that for some constant \( 0 < \varepsilon < 1 \) the sketching matrix \( S \in \mathbb{R}^{d \times s} \) satisfies eqn. (8). Then, for any test data point \( w \in \mathbb{R}^d \), the estimator \( \hat{G} \) returned by Algorithm 1 satisfies

\[ \| (w - m)^T (\hat{G} - G) \|_2 \leq \frac{\varepsilon t}{2\sqrt{\lambda}} \| V V^T (w - m) \|_2. \]

Recall that \( V V^T (w - m) \) is the projection of the vector \( w - m \) onto the row space of \( A \).

**Running time of Algorithm 1.** First, we need to compute \( A \tilde{G}^{(j-1)} \), which takes time \( \mathcal{O}(c \cdot \text{nnz}(A)) \). Then, computing the sketch \( S A \in \mathbb{R}^{n \times s} \) takes \( T(A, S) \) time
which depends on the particular construction of $S$ (see Section 2.2). In order to invert the matrix $\Theta = ASS^T A^T + \lambda I_n$, it suffices to compute the SVD of the matrix $AS$. Notice that given the singular values of $AS$ we can compute the singular values of $\Theta$ and also notice that the left and right singular vectors of $\Theta$ are the same as the left singular vectors of $AS$. Interestingly, we do not need to compute $\Theta^{-1}$: we can store it implicitly by storing its left (and right) singular vectors $U_\Theta$ and its singular values $\Sigma_\Theta$. Then, we can compute all necessary matrix-vector products using this implicit representation of $\Theta^{-1}$. Thus, inverting $\Theta$ takes $O(s n^2)$ time. Updating the matrices $L(I)$, $Y(I)$, and $G(I)$ is dominated by the aforementioned running times. Thus, summing over all $t$ iterations, the running time of Algorithm 1 is
\[ O(t c \cdot \text{nnz}(A)) + O(s n^2) + T(A, S), \] (10)
which should be compared to the $O(n^2 d)$ time that would be needed by standard RFDA approaches.

We note that our algorithm can also be viewed as a preconditioned Richardson iteration with step-size equal to one for solving the linear system $(A A^T + \lambda I_n)F = \Theta$ in $F \in \mathbb{R}^{n \times c}$ with randomized pre-conditioner $P^{-1} = (A SS^T A^T + \lambda I_n)^{-1}$. However, our objective and analysis are significantly different compared to the conventional preconditioned Richardson iteration. First, our matrix of interest is $G = A^T F \in \mathbb{R}^{d \times c}$, whereas standard analysis of the preconditioned Richardson iteration is with respect to $F$. Specifically, in the context of discriminant analysis, for a new observation $w \in \mathbb{R}^d$, we are interested in understanding whether the output of our algorithm closely approximates the original point in the projected space, i.e., if $\| (w - m)^T (G - G) \|_2$ is sufficiently small. To the best of our knowledge, standard analysis of preconditioned Richardson iteration does not yield a bound for $\| (w - m)^T (G - G) \|_2$. Second, our analysis is with respect to the Euclidean norm whereas the standard analysis is in terms of the energy-norm of $(A A^T + \lambda I_n)$, as the matrix $P^{-1}(A A^T + \lambda I_n)$ is not symmetric positive definite.

Finally, we note that our proof also holds when different sampling matrices $S_j$ (for $j = 1, \ldots, t$) are used in each iteration, as long as they satisfy the constraints of eqns. (5) or (8). In fact, the sketching matrices $S_j$ do not even need to have the same number of columns (see Section 5 for an open problem in this setting).

### 2.2 SATYSFYING THE STRUCTURAL CONDITIONS

The structural conditions of eqns. (5) and (8) essentially boil down to randomized, approximate matrix multiplication \cite{11, 12}, a task that has received much attention in the randomized linear algebra community. We discuss general sketching-based approaches here and defer the discussion of sampling-based approaches and the corresponding results to Appendix E. A particularly useful result for our purposes appeared in \cite{9}. Under our notation, \cite{9} proved that for $Z \in \mathbb{R}^{d \times n}$ and for a (suitably constructed) sketching matrix $S \in \mathbb{R}^{d \times \gamma}$, with probability at least $1 - \delta$,
\[ \|Z^T S S^T Z - Z^T Z\|_2 \leq \epsilon \left( |Z|_2^2 + \|z\|_F^2 \right). \] (11)
This bound holds for a broad family of constructions for the sketching matrix $S$ (see \cite{9} for details). In particular, \cite{9} demonstrated a construction for $S$ with $s = O(r/\epsilon^2)$ columns such that, for any $n \times d$ matrix $A$, the product $AS$ can be computed in time $O(n \text{nnz}(A)) + O((r^2 + r^4 n)/\epsilon^2)$ for some constant $\gamma$. In this case, the running time needed to compute the sketch equals $T(A, S) = O(n \text{nnz}(A)) + O(d^2 \gamma n/\epsilon^2)$. The running time of the overall algorithm follows from eqn. (10) and our choices for $s$ and $r$:
\[ O(t c \cdot \text{nnz}(A)) + \tilde{O}(d^2 \gamma n/\epsilon^2 \max(2, \gamma)). \]

The failure probability (hidden in the polylogarithmic terms) can be easily controlled using a union bound. Finally, a simple change of variables (using $\epsilon/4$ instead of $\epsilon$) suffices to satisfy the structural condition of eqn. (5) without changing the above running time.

Similarly, starting with eqn. (8), let $Z = V$ and note that $\|V\|_2^2 = \rho$ and $\|V\|_2 = 1$. Setting $r = \rho$, eqn. (11) implies that $\|V^T S S^T V - I_n\|_2 \leq 2 \epsilon$. In this case, the running time of the sketch computation is equal to $T(A, S) = O(n \text{nnz}(A)) + \tilde{O}(\rho^2 n/\epsilon^2)$. The running time of the overall algorithm follows from eqn. (10) and our choices for $s$ and $r$:
\[ O(t c \cdot \text{nnz}(A)) + \tilde{O}(\rho n^2 / \epsilon^2 \max(2, \gamma)). \]

Again, a simple change of variables suffices to satisfy eqn. (8) without changing the running time.

We note that the above running times can be slightly improved if $s$ is smaller than $n$, since $s$ depends only on the effective degrees of freedom ($d_s$) of the problem (or, on the rank $\rho$ of the data matrix $A$). In this case, the SVD of $AS$ can be computed in $O(n s^2)$ time, and the running time of our algorithm is given by $O(t c \cdot \text{nnz}(A)) + \tilde{O}(d_s^2 n/\epsilon^2 \max(4, \gamma))$ (or, $O(t c \cdot \text{nnz}(A)) + \tilde{O}(\rho n^2 / \epsilon^2 \max(4, \gamma))$).


3 SKETCHING THE PROOF OF THEOREM 1

Due to space considerations, most of our proofs have been deferred to the Appendix. However, to provide a flavor of the mathematical derivations underlying our contributions, we will present an outline of the proof of Theorem 1.

Using the quantities defined in Algorithm 1, let

\[ G^{(j)} = A^T (A A^T + \lambda I_n)^{-1} L^{(j)}, \quad j = 1, \ldots, t. \]  

(12)

Note that \( G = G^{(1)} \). We remind the reader that \( U \in \mathbb{R}^{n \times p}, V \in \mathbb{R}^{d \times p} \) and \( \Sigma \in \mathbb{R}^{p \times p} \) are, respectively, the matrices of the left singular vectors, right singular vectors and singular values of \( A \). We will make extensive use of the matrix \( \Sigma_\lambda \) defined in eqn. (6). The next result provides an alternative expression for \( G^{(j)} \).

Lemma 3. For \( j = 1, \ldots, t \), let \( L^{(j)} \) be the intermediate matrices in Algorithm 1 and let \( G^{(j)} \) be the matrix defined in eqn. (12). Then for any \( j = 1, \ldots, t \), \( G^{(j)} \) can also be expressed as

\[ G^{(j)} = V_\Sigma^2 \Sigma^{-1} U^T L^{(j)}. \]  

(13)

Proof. Using the full SVD representation of \( A \), we have

\[
G^{(j)} = V f \Sigma_f U_f^T (U_f \Sigma_f \Sigma_f^T U_f + \lambda U_f U_f^T)^{-1} L^{(j)}
\]

\[
= V f \Sigma_f^2 (\Sigma_f + \lambda I_f)^{-1} U_f^T L^{(j)}
\]

\[
= (V V_f) \Sigma_f^2 \Sigma_f^{-1} U_f^T L^{(j)}
\]

\[
= (V V_f) \Sigma_f^2 \Sigma_f^{-1} U_f^T L^{(j)}
\]

\[
= (V V_f) \Sigma_f^2 \Sigma_f^{-1} U_f^T L^{(j)}
\]

\[
= (V V_f) \Sigma_f^2 \Sigma_f^{-1} U_f^T L^{(j)}
\]

which completes the proof.

(13)

Our next result (see Appendix C for a detailed proof) provides a bound which later on plays an important role in showing that the underlying error decays exponentially as the number of iterations in Algorithm 1 increases. We state the lemma and outline its proof.

Lemma 4. For \( j = 1, \ldots, t \), let \( L^{(j)} \) be as defined in Algorithm 1 and let \( \tilde{G}^{(j)} \) be defined as in eqn. (12). Further, let \( S \in \mathbb{R}^{d \times s} \) be the sketching matrix and let \( E = \Sigma_\lambda V^T S S^T V \Sigma_\lambda - \Sigma_\lambda^2 \). If eqn. (5) is satisfied, i.e., \( |E|_2 \leq \tilde{\xi} \), then, for all \( j = 1, \ldots, t \),

\[
|| (w - m)^T (G^{(j)} - G^{(j)}) ||_2
\]

\[
\leq \tilde{\xi} || VV^T (w - m) ||_2 || \Sigma_\lambda \Sigma^{-1} U^T L^{(j)} ||_2.
\]

(14)

Proof sketch. Applying Lemma 3 and using the SVD of \( A \) and the fact that \( |E|_2 \leq 1 \), we first express the intermediate matrices \( \tilde{G}^{(j)} \) of Algorithm 1 in terms of the matrices \( G^{(j)} \) of eqn. (12) as

\[ \tilde{G}^{(j)} = G^{(j)} + V \Sigma_\lambda Q \Sigma_\lambda \Sigma^{-1} U^T L^{(j)}, \]

(15)

where \( Q = \sum_{\ell=1}^\infty (-1)\epsilon^\ell E^\ell \). Notice that

\[
||Q||_2 = || \sum_{\ell=1}^\infty (-1)^\ell E^\ell ||_2 \leq \sum_{\ell=1}^\infty ||E||_2
\]

\[
\leq \sum_{\ell=1}^\infty ||E||_2 \leq \sum_{\ell=1}^\infty (\frac{\epsilon}{2})^\ell = \frac{\epsilon/2}{1 - \epsilon/2} \leq \epsilon.
\]

(16)

In the above, we used the triangle inequality, submultiplicativity of the spectral norm, and the fact that \( \epsilon \leq 1 \). Next, we plug-in eqn. (15) and apply submultiplicativity to conclude

\[
|| (w - m)^T (G^{(j)} - G^{(j)}) ||_2
\]

\[
= || (w - m)^T V \Sigma_\lambda Q \Sigma_\lambda \Sigma^{-1} U^T L^{(j)} ||_2
\]

\[
\leq || (w - m)^T V||_2 || \Sigma_\lambda||_2 ||Q||_2 || \Sigma_\lambda \Sigma^{-1} U^T L^{(j)} ||_2
\]

\[
\leq \epsilon ||VV^T (w - m)||_2 || \Sigma_\lambda \Sigma^{-1} U^T L^{(j)} ||_2
\]

where the last inequality follows from eqn. (16) and the fact that \( || \Sigma_\lambda ||_2 \leq 1 \). □

The next lemma presents a structural result for \( G \).

Lemma 5. Let \( \tilde{G}^{(j)}, j = 1, \ldots, t \) be the sequence of matrices introduced in Algorithm 1 and let \( G^{(j)} \in \mathbb{R}^d \) be defined as in eqn. (12). Then, the matrix \( G \) in eqn. (3) can be expressed as

\[ G = G^{(t)} + \sum_{j=1}^{t-1} \tilde{G}^{(j)}. \]

(17)

Proof. We prove the lemma by induction on \( t \). Notice that \( L^{(1)} = \Omega \); thus, for \( t = 1 \), eqn. (12) boils down to

\[ G^{(1)} = A^T (A A^T + \lambda I_n)^{-1} L^{(1)} = G. \]

For \( t = 2 \), we get

\[
G^{(2)} = A^T (A A^T + \lambda I_n)^{-1} L^{(2)}
\]

\[
= A^T (A A^T + \lambda I_n)^{-1} (L^{(1)} - \lambda Y^{(1)} - \lambda G^{(1)})
\]

\[
= A^T (A A^T + \lambda I_n)^{-1} L^{(1)}
\]
\[ -\mathbf{A}^T (\mathbf{ASS}^T \mathbf{A} + \lambda \mathbf{I}_n)^{-1} \mathbf{L}^{(j)} = \mathbf{G} - \tilde{\mathbf{G}}^{(j)}. \]  

(18)

Here, eqn. (18) follows from the fact that \( \mathbf{Y}^{(1)} = (\mathbf{ASS}^T \mathbf{A} + \lambda \mathbf{I}_n)^{-1} \mathbf{L}^{(1)} \). Now, suppose that eqn. (17) is also true for \( t = p \), i.e.,

\[ \mathbf{G}^{(p)} = \mathbf{G} - \sum_{j=1}^{p-1} \tilde{\mathbf{G}}^{(j)}. \]  

(19)

Then, for \( t = p + 1 \), we can express \( \mathbf{G}^{(t)} \) as

\[ \mathbf{G}^{(p+1)} = \mathbf{A}^T (\mathbf{AA}^T + \lambda \mathbf{I}_n)^{-1} \mathbf{L}^{(p+1)} = \mathbf{A}^T (\mathbf{AA}^T + \lambda \mathbf{I}_n)^{-1} (\mathbf{L}^{(p)} - \lambda \mathbf{Y}^{(p)} - \mathbf{A} \tilde{\mathbf{G}}^{(p)}) \]

\[ = \mathbf{A}^T (\mathbf{AA}^T + \lambda \mathbf{I}_n)^{-1} \mathbf{L}^{(p)} - \mathbf{A}^T (\mathbf{ASS}^T \mathbf{A} + \lambda \mathbf{I}_n)^{-1} \mathbf{L}^{(p)} \]

\[ = \mathbf{G}^{(p)} - \tilde{\mathbf{G}}^{(p)} = (\mathbf{G} - \sum_{j=1}^{p-1} \tilde{\mathbf{G}}^{(j)}) - \tilde{\mathbf{G}}^{(p)} = \mathbf{G} - \sum_{j=1}^{p} \tilde{\mathbf{G}}^{(j)}, \]  

(20)

where eqn. (20) holds as \( \mathbf{Y}^{(p)} = (\mathbf{ASS}^T \mathbf{A} + \lambda \mathbf{I}_n)^{-1} \mathbf{L}^{(p)} \). Furthermore, the second last equality follows from eqn. (19). By the induction principle, we have proven eqn. (17). \( \square \)

Repeated application of Lemmas 5 and 4 yields:

\[ \| (\mathbf{w} - \mathbf{m})^T (\tilde{\mathbf{G}} - \mathbf{G})^T \|_2 = \| (\mathbf{w} - \mathbf{m})^T (\sum_{j=1}^{t} \mathbf{G}^{(j)} - \mathbf{G})^T \|_2 \]  

(21)

\[ = \| (\mathbf{w} - \mathbf{m})^T (\tilde{\mathbf{G}}^{(t)} - \mathbf{G} - \sum_{j=1}^{t-1} \tilde{\mathbf{G}}^{(j)})^T \|_2 \]

\[ \leq \| (\mathbf{w} - \mathbf{m})^T (\tilde{\mathbf{G}}^{(t)} - \mathbf{G}^{(t)})^T \|_2 \]

\[ \leq \varepsilon \| \mathbf{V} \mathbf{V}^T (\mathbf{w} - \mathbf{m}) \|_2 \| \Sigma \mathbf{A}^T \mathbf{U}^T \mathbf{L}^{(t)} \|_2. \]

(22)

The next bound (see Appendix C for its detailed proof) provides a critical inequality that can be used recursively in order to establish Theorem 1.

**Lemma 6.** Let \( \mathbf{L}^{(j)} \), \( j = 1, \ldots, t \) be the matrices defined in Algorithm 1. For any \( j = 1, \ldots, t-1 \), if eqn. (5) is satisfied, i.e., \( \| \mathbf{E} \|_2 \leq \varepsilon \), then

\[ \| \Sigma \mathbf{A}^T \mathbf{U}^T \mathbf{L}^{(j+1)} \|_2 \leq \varepsilon \| \Sigma \mathbf{A}^T \mathbf{U}^T \mathbf{L}^{(j)} \|_2. \]

(23)

**Proof sketch.** From Algorithm 1, we have that for \( j = 1, \ldots, t-1 \),

\[ \mathbf{L}^{(j+1)} = \mathbf{L}^{(j)} - \lambda \mathbf{Y}^{(j)} - \mathbf{A} \tilde{\mathbf{G}}^{(j)} \]

\[ = \mathbf{L}^{(j)} - (\mathbf{AA}^T + \lambda \mathbf{I}_n) (\mathbf{ASS}^T \mathbf{A} + \lambda \mathbf{I}_n)^{-1} \mathbf{L}^{(j)}. \]

(24)

Applying the SVD of \( \mathbf{A} \) it can be shown (see Appendix C for details) that

\[ (\mathbf{AA}^T + \lambda \mathbf{I}_n) (\mathbf{ASS}^T \mathbf{A} + \lambda \mathbf{I}_n)^{-1} \mathbf{L}^{(j)} = \mathbf{L}^{(j)} + \mathbf{U}(\Sigma^2 + \lambda \mathbf{I}_p)^{-1} \Sigma \lambda Q \Sigma \lambda^{-1} \mathbf{U}^T \mathbf{L}^{(j)}, \]

(25)

where \( Q = \sum_{l=1}^{\infty} (-1)^l \mathbf{E}^l \).

Combining eqns. (24) and (25), we get

\[ \mathbf{L}^{(j+1)} = -\mathbf{U}(\Sigma^2 + \lambda \mathbf{I}_p)^{-1} \Sigma \lambda Q \Sigma \lambda^{-1} \mathbf{U}^T \mathbf{L}^{(j)}. \]

(26)

Finally, applying eqn. (26), we obtain

\[ \| \Sigma \mathbf{A}^T \mathbf{U}^T \mathbf{L}^{(j+1)} \|_2 \]

\[ = \| \Sigma \mathbf{A}^T \mathbf{U}^T (\Sigma^2 + \lambda \mathbf{I}_p)^{-1} \Sigma \lambda Q \Sigma \lambda^{-1} \mathbf{U}^T \mathbf{L}^{(j)} \|_2 \]

\[ = \| \Sigma \mathbf{A}^T \mathbf{U}^T \mathbf{L}^{(j)} \|_2 \leq \| Q \|_2 \| \Sigma \mathbf{A}^T \mathbf{U}^T \mathbf{L}^{(j)} \|_2 \]

\[ \leq \varepsilon \| \Sigma \mathbf{A}^T \mathbf{U}^T \mathbf{L}^{(j)} \|_2 \]

where the third equality holds since \( \Sigma \mathbf{A}^T \mathbf{U}^T \mathbf{L}^{(j)} = \mathbf{I}_p \). The last two inequalities follow from sub-multiplicativity and the fact that \( \| Q \|_2 \leq \varepsilon \) (by eqn. (16)). \( \square \)

**Proof of Theorem 1.** Applying Lemma 6 iteratively, we obtain

\[ \| \Sigma \mathbf{A}^T \mathbf{U}^T \mathbf{L}^{(t)} \|_2 \leq \varepsilon \| \Sigma \mathbf{A}^T \mathbf{U}^T \mathbf{L}^{(t-1)} \|_2 \]

\[ \leq \ldots \leq \varepsilon^{t-1} \| \Sigma \mathbf{A}^T \mathbf{U}^T \mathbf{L}^{(1)} \|_2. \]

(27)

Notice that \( \mathbf{L}^{(1)} = \mathbf{O} \) by definition. Also, \( \mathbf{O}^T \mathbf{O} = \mathbf{I}_n \) and thus \( \| \mathbf{O} \|_2 = 1 \). Furthermore, we know that \( \| \mathbf{U} \|_2 = 1 \) and \( \| \Sigma \mathbf{A}^T \mathbf{U}^T \mathbf{L}^{(j)} \|_2 = \max_{1 \leq i \leq p} (\sigma_i^2 + \lambda) - \frac{1}{2} \). Thus, sub-multiplicativity yields

\[ \| \Sigma \mathbf{A}^T \mathbf{U}^T \mathbf{L}^{(j)} \|_2 \leq \| \Sigma \mathbf{A}^T \mathbf{U}^T \mathbf{L}^{(j)} \|_2 \]

\[ = \max_{1 \leq i \leq p} (\sigma_i^2 + \lambda) - \frac{1}{2} \leq \lambda - \frac{1}{2}, \]

(28)

where the last inequality holds since \( (\sigma_i^2 + \lambda) - \frac{1}{2} \leq \lambda - \frac{1}{2} \) for all \( i = 1, \ldots, p \).

Finally, combining eqns. (22), (27) and (28), we get

\[ \| (\mathbf{w} - \mathbf{m})^T (\tilde{\mathbf{G}} - \mathbf{G}) \|_2 \leq \frac{\varepsilon t}{\sqrt{\lambda}} \| \mathbf{V} \mathbf{V}^T (\mathbf{w} - \mathbf{m}) \|_2, \]

which concludes the proof. \( \square \)

### 4  EMPIRICAL EVALUATION

**4.1  EXPERIMENT SETUP**

We perform experiments on two real-world datasets: ORL [3] is a database of grey-scale face images with
n = 400 examples and d = 10,304 features, with each example belonging to one of c = 40 classes; PEMS [26] describes the occupancy rate of different car lanes in freeways of the San Francisco bay area, with n = 440 examples, d = 138,672 features, and c = 7 label classes.

In our experiments, we compare both sketching-based and sampling-based constructions for the sketching matrix S. For sketching-based approaches (cf. Section 2.2), we construct S using either the count-sketch matrix [7] as in [32], and the sub-sampled randomized Hadamard transform (SRHT) [1]. For sampling-based approaches (cf. Appendix E), we construct the sampling-and-rescaling matrix S (cf. Algorithm 3 of Appendix E) using three different choices of sampling probabilities: (i) uniformly at random, (ii) proportional to column leverage scores, or (iii) proportional to column ridge leverage scores. Note that constructing S with uniform sampling probabilities do not in general satisfy the structural conditions of eqns. (5) and (8).

For each sketching method, we run Algorithm 1 for 50 iterations with a variety of sketch sizes, and measure the relative approximation error \( \| \hat{G} - G \|_F / \| G \|_F \), where \( \hat{G} \) is computed exactly. We also randomly divide each dataset into a training set with 60% examples and a test set of 40% examples (stratified by label), and measure the classification accuracy on the test set with \( \hat{G} \) estimated from the training set. For each sketching method, we repeat 20 random trials and report the means and standard errors of the experiment results.

4.2 RESULTS AND DISCUSSION

In Figure 1, the first column plots the relative approximation error (for a fixed sketch size) as the iterative algorithm progresses; the second column plots the relative approximation error with respect to varying sketch sizes; and the third column plots the test classification accuracy obtained using the estimated \( \hat{G} = \sum_{j=1}^{t} G^{(j)} \) after \( t = 1, \ldots, 10 \) iterations.

For count-sketch, SRHT, as well as leverage score and ridge leverage score sampling, we observe that the relative approximation error decays exponentially as our iterative algorithm progresses.\(^4\) In particular, constructing the sketching matrix S using the sketching-based approaches appears to yield slightly improved approximation quality over the sampling-based approaches. Furthermore, while leverage score and ridge leverage score sampling perform comparably on the ORL dataset, the latter significantly outperforms the former on the PEMS dataset. This confirms our discussion in Section 1.1: for ridge leverage score sampling, setting \( s = O(\varepsilon^{-2} d_{\lambda} \ln d_{\lambda}) \) suffices to satisfy the structural condition of eqn. (5), while for leverage scores, setting \( s = O(\varepsilon^{-2} \rho \ln \rho) \) suffices to satisfy the structural condition of eqn. (8). (Recall that \( \rho \) can be substantially larger than the effective degrees of freedom \( d_{\lambda} \).) Finally, we note that the proposed approach of [32] (see Theorem 3 therein for the \( d \gg n \) setting) corresponds to running a single iteration of Algorithm 1; our iterative algorithm yields significant improvements in the approximation quality of the solutions.

\(^4\)Except in the last column of Figure 1, we set the regularization parameter to \( \lambda = 10 \) in the RFDA problem as well as the ridge leverage score sampling probabilities.
In the last column of Figure 1, we keep the design matrix unchanged (fixing \(n\)) while varying the regularization parameter \(\lambda\), and plot the relative approximation error against the effective degrees of freedom \(d_\lambda\) of the RFDA problem. We observe that the relative approximation error decreases exponentially as \(d_\lambda\) decreases; thus, the sketch size or number of iterations necessary to achieve a certain approximation precision also decreases with \(d_\lambda\), even though \(n\) remains fixed.

Finally, an exciting open problem would be to investigate whether the use of independent sampling matrices in each iteration of Algorithm 1 (i.e., introducing new “randomness” in each iteration) could lead to provably improved bounds for our main theorems. We conjecture that this is indeed the case, and further experiment results support our conjecture. In particular, Figure 2 plots the relative approximation error vs. number of iterations on the PEMS dataset for various increasing sketch sizes; similar plots for the PEMS dataset are shown in Figure 3 of Appendix G. We observe that using a newly sampled sketching matrix at every iteration enables faster convergence as the iterations progress, and also reduces the sketch size \(s\) necessary for Algorithm 1 to converge.

5 CONCLUSION AND OPEN PROBLEMS

We have presented simple structural results to analyze an iterative, sketching-based RFDA algorithm that guarantees highly accurate solutions compared to conventional approaches. An obvious open problem is to either improve on the sample size requirement of our sketching matrix or present matching lower bounds to show that our bounds are tight. Another open problem would be to explore similar approaches for other versions of regularized FDA that use, say, the pseudo-inverse of the centered data matrix. In addition, unlike the case for sketched ridge regression \([28, 6]\) where the bias–variance trade-off of estimators could be explicitly analyzed, such statistical analyses do not apply to bounding the generalization error of our proposed RFDA algorithm.

Acknowledgements. We thank the anonymous reviewers for their helpful comments. AC and PD were supported by NSF grants IIS-1661760, IIS-1661756, CCF-1814041, and DMS-1760353. JY was supported by NSF grants IIS-1618690, IIS-1546488, and CCF-0939370.

References


Randomized Iterative Algorithms for Fisher Discriminant Analysis
(Appendix)

Appendix A  PRELIMINARIES

We start by reviewing a result regarding the convergence of a matrix von Neumann series for $(I - P)^{-1}$. This will be an important tool in our analysis.

Proposition 7. Let $P$ be any square matrix with $\|P\|_2 < 1$. Then $(I - P)^{-1}$ exists and

$$(I - P)^{-1} = I + \sum_{\ell=1}^{\infty} P^\ell.$$  

Appendix B  EVD-BASED ALGORITHMS FOR FDA

For RFDA, we quote an EVD-based algorithm along with an important result from [36] which together are the building blocks of our iterative framework. Let $M \in \mathbb{R}^{c \times c}$ be the matrix such that $M = \Omega^T A \Gamma$. Clearly, $M$ is symmetric and positive semi-definite.

Algorithm 2  Algorithm for RFDA problem (3)

Input: $A \in \mathbb{R}^{n \times d}$, $\Omega \in \mathbb{R}^{n \times c}$ and $\lambda > 0$;
$G \leftarrow (A^T A + \lambda I_d)^{-1} A^T \Omega$;
$M \leftarrow \Omega^T A \Gamma$;
Compute thin SVD: $M = V_M \Sigma_M V_M^T$.
Output: $X = G V_M$

Theorem 8. Using Algorithm 2, let $X$ be the solution of problem (3), then we have

$$XX^T = GG^T.$$  

For any two data points $w_1, w_2 \in \mathbb{R}^d$, Theorem 8 implies

$$(w_1 - w_2)^T XX^T (w_1 - w_2) = (w_1 - w_2)^T G G^T (w_1 - w_2)$$

$$\iff \| (w_1 - w_2)^T X \|_2 = \| (w_1 - w_2)^T G \|_2.$$  

Theorem 8 indicates that if we use any distance-based classification method such as $k$-nearest neighbors, both $X$ and $G$ shares the same property. Thus, we may shift our interest from $X$ to $G$.

Appendix C  PROOF OF THEOREM 1

Proof of Lemma 3. Using the full SVD representation of $A$ we have

$$G^{(j)} = V_j \Sigma_j^T U_j^T (U_j \Sigma_j \Sigma_j^T U_j \Gamma + \lambda U_j U_j^T)^{-1} L^{(j)}$$

$$= V_j \Sigma_j^T (\Sigma_j \Sigma_j^T + \lambda I_n)^{-1} U_j^T L^{(j)}$$

$$= (V \ V_\perp) \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \Sigma^2 + \lambda I_{\rho} & 0 \\ 0 & \lambda I_{n-\rho} \end{pmatrix}^{-1} \begin{pmatrix} U^T \\ U_\perp \end{pmatrix} L^{(j)}$$

$$= (V \ V_\perp) \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \Sigma^2 + \lambda I_{\rho} & 0 \\ 0 & \lambda I_{n-\rho} \end{pmatrix}^{-1} \begin{pmatrix} U^T \\ U_\perp \end{pmatrix} L^{(j)}$$

$$= (V \ V_\perp) \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \Sigma^2 + \lambda I_{\rho} & 0 \\ 0 & \lambda I_{n-\rho} \end{pmatrix}^{-1} \begin{pmatrix} U^T \\ U_\perp \end{pmatrix} L^{(j)}$$
\begin{align*}
= (V, V_\perp) \left( \Sigma (\Sigma^2 + \lambda I_\rho)^{-1} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right) \left( \begin{pmatrix} U_T^T \\ U_\perp^T \end{pmatrix} \right) L^{(j)}
= V \Sigma (\Sigma^2 + \lambda I_\rho)^{-1} U_T^T L^{(j)}
= V \Sigma \Sigma^{-1} (I_\rho + \lambda \Sigma^{-2})^{-1} \Sigma^{-1} U_T^T L^{(j)}
= V \Sigma^2 \Sigma^{-1} U_T^T L^{(j)},
\end{align*}
\tag{29}

which completes the proof. \hfill \Box

**Detailed proof of Lemma 4.** First, using SVD of \(A\), we express \(\tilde{G}^{(j)}\) in terms of \(G^{(j)}\).

\[\tilde{G}^{(j)} = V_j \Sigma_j^T U_j^T (U_j \Sigma_j V_j^T \Sigma_j V_j^T U_j^T + \lambda U_j U_j^T)^{-1} L^{(j)} = V_j \Sigma_j^T (\Sigma_j V_j^T \Sigma V_j^T + \lambda I_n)^{-1} U_j L^{(j)} \]

\[= (V, V_\perp) \left( \Sigma 0 \vline 0 \right) \left( \begin{pmatrix} \Sigma V^T SS^T V \Sigma & 0 \\ 0 & 0 \end{pmatrix} + \lambda I_n \right)^{-1} \left( \begin{pmatrix} U_T^T \\ U_\perp^T \end{pmatrix} \right) L^{(j)} \]

\[= (V, V_\perp) \left( \begin{pmatrix} \Sigma 0 \vline 0 \end{pmatrix} \left( \begin{pmatrix} \Sigma V^T SS^T V \Sigma & 0 \\ 0 & 0 \end{pmatrix} + \lambda I_n \right)^{-1} \left( \begin{pmatrix} U_T^T \\ U_\perp^T \end{pmatrix} \right) L^{(j)} \]

\[= (V, V_\perp) \left( \begin{pmatrix} \Sigma 0 \vline 0 \end{pmatrix} \left( (\Sigma V^T SS^T V \Sigma + \lambda I_{\rho})^{-1} 0 \\ 0 \vline \frac{1}{\lambda} I_{n-\rho} \right) \left( \begin{pmatrix} U_T^T \\ U_\perp^T \end{pmatrix} \right) L^{(j)} \]

\[= V \Sigma (\Sigma^2 + \lambda \Sigma \Sigma^{-2})^{-1} U_T L^{(j)} \tag{30} \]

\[= V \Sigma (\Sigma^2 + \lambda \Sigma \Sigma^{-2})^{-1} U_T L^{(j)} \tag{31} \]

\[= V \Sigma (\Sigma^2 + \lambda \Sigma \Sigma^{-2})^{-1} U_T L^{(j)} \tag{32} \]

Eqn. (31) used the fact that \(\Sigma_j V^T SS^T V \Sigma_j = \Sigma^2 + \Sigma\). Eqn. (32) follows from the fact that \(\Sigma^2 + \lambda \Sigma \Sigma^{-2} \Sigma \in \mathbb{R}^{n \times n}\) is a diagonal matrix with \(i\)-th diagonal element

\[\left( \Sigma^2 + \lambda \Sigma \Sigma^{-2} \Sigma \right)_ii = \frac{\sigma_i^2}{\sigma_i^2 + \lambda} + \frac{\lambda}{\sigma_i^2 + \lambda} = 1,\]

for any \(i = 1 \ldots \rho\). Thus, we have \((\Sigma^2 + \lambda \Sigma \Sigma^{-2} \Sigma) = I_\rho\). Since \(\|E\|_2 < 1\), Proposition 7 implies that \((I_\rho + E)^{-1}\) exists and

\[(I_\rho + E)^{-1} = I_\rho + \sum_{\ell=1}^{\infty} (-1)^\ell E' = I_\rho + Q.\]

Thus, eqn. (32) can further be expressed as

\[\tilde{G}^{(j)} = V \Sigma \Sigma^{-1} (I_\rho + E)^{-1} \Sigma \Sigma^{-1} U_T L^{(j)} = V \Sigma \Sigma^{-1} (I_\rho + Q) \Sigma \Sigma^{-1} U_T L^{(j)} = V \Sigma^2 \Sigma^{-1} U_T L^{(j)} + V \Sigma Q \Sigma \Sigma^{-1} U_T L^{(j)} = G^{(j)} + V \Sigma Q \Sigma \Sigma^{-1} U_T L^{(j)},\]

where the last line follows from Lemma 3. Further, we have

\[\|Q\|_2 = \| \sum_{\ell=1}^{\infty} (-1)^\ell E'\|_2 \leq \sum_{\ell=1}^{\infty} \|E'\|_2 \leq \sum_{\ell=1}^{\infty} \|E\|_2 = \sum_{\ell=1}^{\infty} \left( \frac{\varepsilon}{2} \right)^{\ell} = \frac{\varepsilon/2}{1 - \varepsilon/2} \leq \varepsilon,\]

\[\tag{34}\]
where we used the triangle inequality, the sub-multiplicativity of the spectral norm, and the fact that \( \varepsilon \leq 1 \). Next, we combine eqns. (33) and (34) to get
\[
\|(w - m)^T (G^{(j)} - G^{(j)})\|_2 = \|(w - m)^T \Sigma \Sigma^{-1} U^T L^{(j)}\|_2
\]
\[
\leq \|(w - m)^T \Sigma \|_2 \|Q\|_2 \|\Sigma \Sigma^{-1} U^T L^{(j)}\|_2
\]
\[
\leq \varepsilon \|V V^T (w - m)\|_2 \|\Sigma \Sigma^{-1} U^T L^{(j)}\|_2,
\]
which completes the proof.

The next bound provides a critical inequality that can be used recursively to establish Theorem 1.

**Detailed proof of Lemma 6.** From Algorithm 1, we have for \( j = 1 \ldots t - 1 \)
\[
L^{(j+1)} = L^{(j)} - \lambda Y^{(j)} - A \tilde{G}^{(j)}
\]
\[
= L^{(j)} - (A A^T + \lambda I_n)(A S S^T + \lambda I_n)^{-1} L^{(j)}.
\]
(36)

Now, starting with the full SVD of \( A \), we get
\[
(\lambda I_n)(A S S^T + \lambda I_n)^{-1} L^{(j)}
\]
\[
= (U \Sigma U^T + \lambda I_n)(A S S^T + \lambda I_n)^{-1} L^{(j)}
\]
\[
= U (\Sigma \Sigma^{-1} U^T + \lambda U U^T) (U \Sigma U^T + \lambda U U^T)^{-1} L^{(j)}
\]
\[
= U (\Sigma \Sigma^{-1} U^T + \lambda n I_n) U^T L^{(j)}
\]
\[
= U (\Sigma + \lambda \rho) (\Sigma + \lambda \rho)^{-1} L^{(j)}
\]
\[
= U (\Sigma + \lambda \rho)^{-1} L^{(j)}
\]
\[
= U (\Sigma + \lambda \rho)^{-1} L^{(j)}
\]
\[
= U (\Sigma + \lambda \rho)^{-1} U^T L^{(j)}
\]
\[
= U (\Sigma + \lambda \rho)^{-1} U^T L^{(j)}
\]
\[
= U (\Sigma + \lambda \rho)^{-1} U^T L^{(j)}
\]
(37)

Here, eqn. (38) holds because \( \Sigma \lambda V^T S S^T V \Sigma \lambda = \Sigma \lambda^2 + E \) and the fact that \( \Sigma \lambda^2 + \lambda \Sigma \lambda^2 - \Sigma \lambda^2 \in \mathbb{R}^{n \times n} \) is a diagonal matrix whose \( i \)th diagonal element satisfies
\[
(\Sigma \lambda^2 + \lambda \Sigma \lambda^2 - \Sigma \lambda^2)_{ii} = \frac{\sigma_i^2}{\sigma_i^2 + \lambda} + \lambda \frac{1}{\sigma_i^2 + \lambda} = 1,
\]
for any \( i = 1 \ldots \rho \). Thus, we have \( \Sigma \lambda^2 + \lambda \Sigma \lambda^2 - \Sigma \lambda^2 = I_\rho \). Since \( \|E\|_2 < 1 \), Proposition 7 implies that \( (I_\rho + E)^{-1} \) exists and
\[
(I_\rho + E)^{-1} = I_\rho + \sum_{\ell=1}^{\infty} (-1)^\ell E^\ell = I_\rho + Q,
\]
where \( Q = \sum_{\ell=1}^{\infty} (-1)^\ell E^\ell \).

Thus, we rewrite eqn. (38) as
\[
(A A^T + \lambda I_n)(A S S^T + \lambda I_n)^{-1} L^{(j)}
\]
\[ = U \Sigma \Sigma^{-1} U^T L(j) + U(\Sigma^2 + \lambda I_p) \Sigma^{-1} \Sigma \Lambda (I_p + E)^{-1} \Sigma \Sigma^{-1} U^T L(j) \]
\[ = U \Sigma \Sigma^{-1} U^T L(j) + U(\Sigma^2 + \lambda I_p) \Sigma^{-1} \Sigma \Lambda (I_p + Q) \Sigma \Lambda \Sigma^{-1} U^T L(j) \]
\[ = U \Sigma \Sigma^{-1} U^T L(j) + U(\Sigma^2 + \lambda I_p) \Sigma^{-1} \Sigma \Lambda \Sigma^{-1} U^T L(j) + U(\Sigma^2 + \lambda I_p) \Sigma^{-1} \Sigma \Lambda Q \Sigma \Lambda \Sigma^{-1} U^T L(j) \]
\[ = U \Sigma \Sigma^{-1} U^T L(j) + U(\Sigma^2 + \lambda I_p) \Sigma^{-1} \Sigma \Lambda \Sigma^{-1} U^T L(j) \]
\[ = U \Sigma \Sigma^{-1} U^T L(j) + U(\Sigma^2 + \lambda I_p) \Sigma^{-1} \Sigma \Lambda \Sigma^{-1} U^T L(j) \]
\[ = U \Sigma \Sigma^{-1} U^T L(j) + U(\Sigma^2 + \lambda I_p) \Sigma^{-1} \Sigma \Lambda \Sigma^{-1} U^T L(j) \]

Eqn. (39) holds as \((\Sigma^2 + \lambda I_p) \Sigma^{-1} \Sigma \Lambda = I_p\). Further, using the fact that \(U_j U_j^T = I_n\), we rewrite eqn. (40) as
\[ (A A^T + \lambda I_n)(A S S^T A^T + \lambda I_n)^{-1} L(j) = \Sigma \Sigma^{-1} \Sigma \Lambda \Sigma^{-1} U^T L(j) \]

Thus, combining eqns. (36) and (41), we have
\[ L(j+1) = -U(\Sigma^2 + \lambda I_p) \Sigma^{-1} \Sigma \Lambda \Sigma^{-1} U^T L(j) \]

Finally, using eqn. (42), we obtain
\[ \| \Sigma \Lambda \Sigma^{-1} U^T L(j+1) \|_2 = \| \Sigma \Lambda \Sigma^{-1} U^T U(\Sigma^2 + \lambda I_p) \Sigma^{-1} \Sigma \Lambda Q \Sigma \Lambda \Sigma^{-1} U^T L(j) \|_2 \]
\[ = \| \Sigma \Lambda \Sigma^{-1} (\Sigma^2 + \lambda I_p) \Sigma^{-1} \Sigma \Lambda Q \Sigma \Lambda \Sigma^{-1} U^T L(j) \|_2 \]
\[ = \| Q \Sigma \Lambda^{-1} U^T L(j) \|_2 \leq \| Q \|_2 \| \Sigma \Lambda^{-1} U^T L(j) \|_2 \]
\[ \leq \varepsilon \| \Sigma \Lambda^{-1} U^T L(j) \|_2 \]

where the third equality holds as \(\Sigma \Lambda \Sigma^{-1} = I_p\) and the last two steps follow from sub-multiplicativity and eqn. (34) respectively. This concludes the proof.

**Proof of Theorem 1.** Applying Lemma 6 iteratively, we get
\[ \| \Sigma \Lambda \Sigma^{-1} U^T L(f) \|_2 \leq \varepsilon \| \Sigma \Lambda \Sigma^{-1} U^T L(f-1) \|_2 \leq \ldots \leq \varepsilon^{t-1} \| \Sigma \Lambda \Sigma^{-1} U^T L(1) \|_2. \]  

Now, from eqn (43), we apply sub-multiplicativity to obtain
\[ \| \Sigma \Lambda \Sigma^{-1} U^T L(t) \|_2 \leq \| \Sigma \Lambda \Sigma^{-1} \|_2 \| U^T \|_2 \| \Omega \|_2 \leq \max_{1 \leq i \leq p} (\sigma_i^2 + \lambda)^{-\frac{1}{2}} \leq \lambda^{-\frac{1}{2}}, \]

Notice that \(L(1) = \Omega\) by definition. Also, \(\Omega^T \Omega = I_c\) and thus \(\| \Omega \|_2 = 1\). Furthermore, we know that \(\| U^T \|_2 = 1\) and \(\| \Sigma \Lambda \Sigma^{-1} \|_2 = \max_{1 \leq i \leq p} (\sigma_i^2 + \lambda)^{-\frac{1}{2}}\) and the last inequality holds since \((\sigma_i^2 + \lambda)^{-\frac{1}{2}} \leq \lambda^{-\frac{1}{2}}\) for all \(i = 1 \ldots p\).

Finally, combining eqns. (22), (43) and (44), we conclude
\[ \| (w - m)^T (G - H) \|_2 \leq \frac{\varepsilon t}{\sqrt{\lambda}} \| V V^T (w - m) \|_2, \]
which completes the proof.

**Appendix D  PROOF OF THEOREM 2**

**Lemma 9.** For \(j = 1 \ldots t\), let \(L(j)\) and \(G(j)\) be the intermediate matrices in Algorithm 1, \(G(j)\) be the matrix defined in eqn. (12) and \(R\) be defined as in Lemma 3. Further, let \(S \in \mathbb{R}^{d \times s}\) be the sketching matrix and define \(E = V^T S S^T V - I_p\). If eqn. (8) is satisfied, i.e., \(\| E \|_2 \leq \frac{\varepsilon}{2}\), then for all \(j = 1, \ldots, t\), we have
\[ \| (w - m)^T (G(j) - G(j)) \|_2 \leq \varepsilon \| V V^T (w - m) \|_2 \| R^{-1} \Sigma^{-1} U^T L(j) \|_2, \]  

where \(R = I_p + \lambda \Sigma^{-2}\).
Proof. Note that $\Sigma_\Lambda^2 = R^{-1}$. Applying Lemma 3, we can express $G^{(j)}$ as
\begin{equation}
G^{(j)} = V R^{-1} \Sigma^{-1} U^T L^{(j)}. \tag{46}
\end{equation}
Next, rewriting eqn. (30) gives
\begin{equation}
\tilde{G}^{(j)} = V \Sigma (\Sigma V^T S S^T V \Sigma + \Lambda I_p)^{-1} U^T L^{(j)}
= V \Sigma (\Sigma (I_p + \hat{E}) \Sigma + \Lambda I_p)^{-1} U^T L^{(j)} = V \Sigma \Sigma^{-1} (I_p + \hat{E} + \Lambda \Sigma^{-2})^{-1} \Sigma^{-1} U^T L^{(j)}
= V (R + \hat{E})^{-1} \Sigma^{-1} U^T L^{(j)} = V (R (I_p + R^{-1} \hat{E}))^{-1} \Sigma^{-1} U^T L^{(j)}. \tag{47}
\end{equation}
Further, notice that
\begin{equation}
\|R^{-1} \hat{E}\|_2 \leq \|R^{-1}\|_2 \|\hat{E}\|_2 \leq \|R^{-1}\|_2 \frac{\varepsilon}{2} = \left(\frac{\sigma_1^2}{\sigma_1^2 + \lambda}\right) \frac{\varepsilon}{2} \leq \frac{\varepsilon}{2} < 1. \tag{49}
\end{equation}
Now, Proposition 7 implies that $(I_p + R^{-1} \hat{E})^{-1}$ exists. Let $\hat{Q} = \sum_{\ell=1}^\infty (-1)^\ell (R^{-1} \hat{E})^\ell$, we have
\begin{equation}
(I_p + R^{-1} \hat{E})^{-1} = I_p + \sum_{\ell=1}^\infty (-1)^\ell (R^{-1} \hat{E})^\ell = I_p + \hat{Q}. \tag{50}
\end{equation}
Thus, we can rewrite eqn. (48) as
\begin{equation}
\tilde{G}^{(j)} = V (I_p + \hat{Q}) R^{-1} \Sigma^{-1} U^T L^{(j)}
= V R^{-1} \Sigma^{-1} U^T L^{(j)} + V \hat{Q} R^{-1} \Sigma^{-1} U^T L^{(j)}
= G^{(j)} + V \hat{Q} R^{-1} \Sigma^{-1} U^T L^{(j)}, \tag{51}
\end{equation}
where eqn. (50) follows eqn. (46). Further, using eqn. (49), we have
\begin{equation}
\|\hat{Q}\|_2 = \|\sum_{\ell=1}^\infty (-1)^\ell (R^{-1} \hat{E})^\ell\|_2 \leq \sum_{\ell=1}^\infty \| (R^{-1} \hat{E})^\ell\|_2 \leq \sum_{\ell=1}^\infty \|R^{-1} \hat{E}\|_2 \leq \sum_{\ell=1}^\infty \left(\frac{\varepsilon}{2}\right)^\ell = \frac{\varepsilon/2}{1 - \varepsilon/2} \leq \varepsilon, \tag{51}
\end{equation}
where we used the triangle inequality, sub-multiplicativity of the spectral norm, and the fact that $\varepsilon \leq 1$. Next, we combine eqns. (50) and (51) to get
\begin{equation}
\|(w - m)^T (\tilde{G}^{(j)} - G^{(j)})\|_2 = \|(w - m)^T V \hat{Q} R^{-1} \Sigma^{-1} U^T L^{(j)}\|_2 
\leq \|(w - m)^T V\|_2 \|\hat{Q}\|_2 \|R^{-1} \Sigma^{-1} U^T L^{(j)}\|_2 
\leq \varepsilon \|(w - m)^T V\|_2 \|R^{-1} \Sigma^{-1} U^T L^{(j)}\|_2 
= \varepsilon \|VV^T (w - m)\|_2 \|R^{-1} \Sigma^{-1} U^T L^{(j)}\|_2, \tag{52}
\end{equation}
where the first inequality follows from sub-multiplicativity and the second last equality holds due to the unitary invariance of the spectral norm. This concludes the proof. \hfill \Box

Remark 10. Repeated application of Lemmas 5 and 9 yields:
\begin{equation}
\|(w - m)^T (\tilde{G} - G)\|_2 = \|(w - m)^T (\sum_{j=1}^{t} G^{(j)} - G)\|_2 = \|(w - m)^T (\tilde{G}^{(t)} - (G - \sum_{j=1}^{t-1} G^{(j)}))\|_2 
= \|(w - m)^T (G^{(t)} - G^{(t)})\|_2 \leq \varepsilon \|VV^T (w - m)\|_2 \|R^{-1} \Sigma^{-1} U^T L^{(t)}\|_2. \tag{53}
\end{equation}

The next bound provides a critical inequality that can be used recursively in order to establish Theorem 2.

Lemma 11. Let $L^{(j)}$, $j = 1, \ldots, t$, be the matrices of Algorithm 1 and $R$ is as defined in Lemma 3. For any $j = 1, \ldots, t - 1$, define $\tilde{E} = V^T S S^T V - I_p$. If eqn. (8) is satisfied i.e. $\|\tilde{E}\|_2 \leq \frac{\varepsilon}{2}$, then
\begin{equation}
\|R^{-1} \Sigma^{-1} U^T L^{(j+1)}\|_2 \leq \varepsilon \|R^{-1} \Sigma^{-1} U^T L^{(j)}\|_2. \tag{54}
\end{equation}
Proof. From Algorithm 1, we have for $j = 1, \ldots, t - 1$,
\[ L^{(j+1)} = L^{(j)} - \lambda Y^{(j)} - A \tilde{G}^{(j)} = L^{(j)} - (A A^T + \lambda I_n)(A S S^T A^T + \lambda I_n)^{-1}L^{(j)}. \tag{55} \]

Rewriting eqn. (37), we have
\[
\begin{align*}
(A A^T + \lambda I_n)(A S S^T A^T + \lambda I_n)^{-1}L^{(j)} & = U^T U^T(j) + U^T(\Sigma^2 + \lambda I_p)(\Sigma^T S S^T \Sigma + \lambda I_p)^{-1}U^T(j) \\
& = U^T U^T(j) + U^T(\Sigma^2 + \lambda I_p)(\Sigma(I_ho + \tilde{E}) + \lambda I_p)^{-1}U^T(j) \\
& = U^T U^T(j) + U^T(\Sigma^2 + \lambda I_p)\Sigma^{-1}(I_ho + \tilde{E})^{-1}U^T(j).
\end{align*}
\]  

Here, eqn. (56) holds because $(I_ho + \tilde{E}) + \lambda \Sigma^{-2}$ is invertible since it is a positive definite matrix. In addition, using the fact that $R = (I_ho + \Sigma^{-2})$, we rewrite eqn. (56) as
\[
\begin{align*}
(A A^T + \lambda I_n)(A S S^T A^T + \lambda I_n)^{-1}L^{(j)} & = U^T U^T(j) + U^T(\Sigma^2 + \lambda I_p)\Sigma^{-1}(R + \tilde{E})^{-1} \Sigma^{-1}U^T(j) \\
& = U^T U^T(j) + U^T(\Sigma^2 + \lambda I_p)\Sigma^{-1}(R(I_ho + R\tilde{E})^{-1}(R - \tilde{E})^{-1} \Sigma^{-1}U^T(j) \\
& = U^T U^T(j) + U^T(\Sigma^2 + \lambda I_p)\Sigma^{-1}(I_ho + \tilde{E})^{-1}(R - \tilde{E})^{-1} \Sigma^{-1}U^T(j) \\
& = U^T U^T(j) + U^T(\Sigma^2 + \lambda I_p)\Sigma^{-1}(I_ho + \tilde{E})^{-1}(R - \tilde{E})^{-1} \Sigma^{-1}U^T(j) \\
& = U^T U^T(j) + U^T(\Sigma^2 + \lambda I_p)\Sigma^{-1}(I_ho + \tilde{E})^{-1}(R - \tilde{E})^{-1} \Sigma^{-1}U^T(j) \\
& = U^T U^T(j) + U^T(\Sigma^2 + \lambda I_p)\Sigma^{-1}(I_ho + \tilde{E})^{-1}(R - \tilde{E})^{-1} \Sigma^{-1}U^T(j) \\
& = U^T U^T(j) + U^T(\Sigma^2 + \lambda I_p)\Sigma^{-1}(I_ho + \tilde{E})^{-1}(R - \tilde{E})^{-1} \Sigma^{-1}U^T(j).
\end{align*}
\]  

The second and third equalities follow from Proposition 7 (using eqn. (49)) and the fact that $R^{-1}$ exists. Further, $Q$ is as defined as in Lemma 9. Moreover, the second last equality holds as $(\Sigma^2 + \lambda I_p)\Sigma^{-1}R^{-1} \Sigma^{-1} = I_p$. Now, using the fact that $U^T U^T = I_n$, we rewrite eqn. (57) as
\[
\begin{align*}
(A A^T + \lambda I_n)(A S S^T A^T + \lambda I_n)^{-1}L^{(j)} & = L^{(j)} + U^T(\Sigma^2 + \lambda I_p)\Sigma^{-1}(R - \tilde{E})^{-1} \Sigma^{-1}U^T(j) \\
& = L^{(j)} + U^T(\Sigma^2 + \lambda I_p)\Sigma^{-1}(R - \tilde{E})^{-1} \Sigma^{-1}U^T(j) \\
& = L^{(j)} + U^T(\Sigma^2 + \lambda I_p)\Sigma^{-1}(R - \tilde{E})^{-1} \Sigma^{-1}U^T(j) \\
& = L^{(j)} + U^T(\Sigma^2 + \lambda I_p)\Sigma^{-1}(R - \tilde{E})^{-1} \Sigma^{-1}U^T(j).
\end{align*}
\]  

Thus, combining, eqns. (55) and (58), we have
\[
L^{(j+1)} = - U^T(\Sigma^2 + \lambda I_p)\Sigma^{-1}(R - \tilde{E})^{-1} \Sigma^{-1}U^T(j). \tag{59}
\]

Finally, from eqn. (59), we obtain
\[
\begin{align*}
\|R^{-1} \Sigma^{-1}U^T(j+1)\|_2 & = \|R^{-1} \Sigma^{-1}U^T(\Sigma^2 + \lambda I_p)\Sigma^{-1}(R - \tilde{E})^{-1} \Sigma^{-1}U^T(j)\|_2 \\
& = \|R^{-1} \Sigma^{-1}(\Sigma^2 + \lambda I_p)\Sigma^{-1}(R - \tilde{E})^{-1} \Sigma^{-1}U^T(j)\|_2 \\
& = \|Q^{-1} \Sigma^{-1}U^T(j)\|_2 \leq \|Q\|_2 \|R^{-1} \Sigma^{-1}U^T(j)\|_2 \\
& \leq \|R^{-1} \Sigma^{-1}U^T(j)\|_2,
\end{align*}
\]  

where the third inequality holds as $R^{-1} \Sigma^{-1}(\Sigma^2 + \lambda I_p)\Sigma^{-1} = I_p$ and the last two steps follow from sub-multiplicativity and eqn. (51) respectively. This concludes the proof.

Proof of Theorem 2. Applying Lemma 11 iteratively, we have
\[
\|R^{-1} \Sigma^{-1}U^T(j)\|_2 \leq \epsilon \|R^{-1} \Sigma^{-1}U^T(j-1)\|_2 \leq \ldots \leq \epsilon^{j-1} \|R^{-1} \Sigma^{-1}U^T(1)\|_2. \tag{61}
\]

Now, from eqn (61) and noticing that $L^{(1)} = \Omega$ by definition, we have
\[
\|R^{-1} \Sigma^{-1}U^T(1)\|_2 \leq \|R^{-1} \Sigma^{-1}\|_2 \|U^T\|_2 \|\Omega\|_2 = \max_{1 \leq i \leq \rho} \left\{ \frac{\sigma_i}{\sigma_i^2 + \lambda} \right\} \leq \frac{1}{2\sqrt{\lambda}}, \tag{62}
\]
where we used sub-multiplicativity and the facts that $\|U^T\|_2 = 1$, $\Omega^T\Omega = I_c$, and $\|\Omega\|_2 = 1$. The last step in eqn. (62) holds since for all $i = 1 \ldots \rho$,

$$
(s_i - \sqrt{\lambda})^2 \geq 0 \implies \sigma_i^2 + \lambda \geq 2s_i\sqrt{\lambda} \implies \frac{\sigma_i}{\sigma_i^2 + \lambda} \leq \frac{1}{2\sqrt{\lambda}}.
$$

Finally, combining eqns. (53), (61) and (62), we obtain

$$
\|(w - m)^T(\hat{G} - G)\|_2 \leq \frac{\varepsilon}{2\sqrt{\lambda}} \|\Sigma V\|_2 (w - m)\|_2,
$$

which concludes the proof.

\[\square\]

### Appendix E  SAMPLING-BASED CONSTRUCTIONS

We now discuss how to satisfy the conditions of eqns. (5) or (8) by sampling, i.e., selecting a small number of features. Towards that end, consider Algorithm 3 for the construction of the sampling-and-rescaling matrix $S$. Finally, the next result appeared in [6] as Theorem 3 and is a strengthening of Theorem 4.2 of [20], since the sampling complexity $s$ is improved to depend only on $\|Z\|_F^2$ instead of the stable rank of $Z$ when $\|Z\|_2 \leq 1$. We also note that Lemma 12 is implicit in [8].

**Algorithm 3 Sampling-and-rescaling matrix**

**Input:** Sampling probabilities $p_i$, $i = 1, \ldots, d$; number of sampled columns $s \ll d$; 

$S \leftarrow 0_{d \times s}$; 

for $t = 1$ to $s$ do 

Pick $i_t \in \{1, \ldots, d\}$ with $P(i_t = i) = p_i$; 

$S_{i_t} = 1/\sqrt{s}p_i$; 

end for 

**Output:** Return $S$;

**Lemma 12.** Let $Z \in \mathbb{R}^{d \times n}$ with $\|Z\|_2 \leq 1$ and let $S$ be constructed by Algorithm 3 with

$$s \geq \frac{8\|Z\|_F^2}{3\varepsilon^2} \ln \left( \frac{4(1 + \|Z\|_F^2)}{\delta} \right),$$

then, with probability at least $1 - \delta$,

$$\|Z^TSS^TZ - Z^TZ\|_2 \leq \varepsilon.$$

Applying Lemma 12 with $Z = V\Sigma$, we can satisfy the condition of eqn. (5) using the sampling probabilities $p_i = \|(V\Sigma)_{\ast i}\|_2^2/d_\lambda$ (recall that $\|V\Sigma_{\ast i}\|_F^2 = d_\lambda$ and $\|V\Sigma\|_2 \leq 1$). It is easy to see that these probabilities are exactly proportional to the column ridge leverage scores of the design matrix $A$. Setting $s = \mathcal{O}(\varepsilon^{-2}d_\lambda \ln d_\lambda)$ suffices to satisfy the condition of eqn. (5). We note that approximate ridge leverage scores also suffice and that their computation can be done efficiently without computing $V$ [8]. Finally, applying Lemma 12 with $Z = V$ we can satisfy the condition of eqn. (8) by simply using the sampling probabilities $p_i = \|V_{\ast i}\|_2^2/\rho$ (recall that $\|V\|_F^2 = \rho$ and $\|V\|_2 = 1$), which correspond to the column leverage scores of the design matrix $A$. Setting $s = \mathcal{O}(\varepsilon^{-2} \rho \ln \rho)$ suffices to satisfy the condition of eqn. (8). We note that approximate leverage scores also suffice and that their computation can be done efficiently without computing $V$ [13].

### Appendix F  SKETCH-SIZE REQUIREMENTS FOR STRUCTURAL CONDITIONS

We provide details on the sketch-size requirements for satisfying the structural conditions of eqns. (5) or (8) when various constructions of the sketching matrix $S$ are used. It was shown in [9] that eqn. (11) can be achieved using a count-sketch matrix $S$ with $s = \mathcal{O}(\frac{r}{\delta^2})$ columns or an SRHT matrix $S$ with $s = \mathcal{O}(\varepsilon^{-2}(r + \log(1/\delta)) \log(\varepsilon))$ columns (here, $\delta$ is the failure probability). As discussed in Section 2.2, setting $r = d_\lambda$ or $r = \rho$ in eqn. (11) for eqns. (5) or (8), respectively, we obtain the sketch-size requirements summarized in Table 1.

### Appendix G  ADDITIONAL EXPERIMENT RESULTS

Table 2 shows the CPU wall-clock times for running RFDA (on a single-core Intel Xeon E5-2660 CPU at 2.6GHz) by either computing $G$ exactly in eqn. (3) or via our iterative algorithm. For both datasets, we report the per-iteration runtime of our algorithm with various sketching-matrix constructions using a sketch size of $s = 5,000$. 
As noted in Section 5, we conjecture that using independent sampling matrices in each iteration of Algorithm 1 (i.e., introducing new “randomness” in each iteration) could lead to improved bounds for our main theorems. We evaluate this conjecture empirically by comparing the performance of Algorithm 1 using either a single sketching matrix \( S \) (the setup in the main paper) or sampling (independently) a new sketching matrix at every iteration \( j \).

Figure 3 shows the relative approximation error vs. number of iterations on the PEMS dataset for increasing sketch sizes. Figure 4 plots the relative approximation error vs. sketch size after 10 iterations of Algorithm 1 were run. We observe that using a newly sampled sketching matrix at every iteration enables faster convergence as the iterations progress, and also reduces the sketch size \( s \) necessary for Algorithm 1 to converge.

![Figure 3](image-url) Relative approximation error (on log-scale) vs. number of iterations on PEMS dataset for increasing sketch size \( s \). Top row: using a single sketching matrix \( S \) throughout. Bottom row: sample a new \( S_j \) at every iteration \( j \).

![Figure 4](image-url) Relative approximation error vs. sketch size on ORL and PEMS after 10 iterations. Single \( S \): using a single sketching matrix \( S \) throughout the iterations. Multiple \( S_j \): sample a new \( S_j \) at every iteration \( j \). Errors are on log-scale; note the difference in magnitude of the approximation errors across plots.

### Table 1: Sketch-size requirements for satisfying eqns. (5) or (8) with probability at least \( 1 - \delta \).

<table>
<thead>
<tr>
<th>Eqn. (5) ( s = \mathcal{O}\left( \frac{d \log (1/\epsilon)}{\epsilon^2} \right) )</th>
<th>Eqn. (8) ( s = \mathcal{O}\left( \frac{\rho + \log (1/\epsilon)}{\epsilon^2} \log \frac{\rho}{\epsilon} \right) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \rho )</td>
<td>( \rho )</td>
</tr>
<tr>
<td>( d )</td>
<td>( d )</td>
</tr>
<tr>
<td>( \epsilon )</td>
<td>( \epsilon )</td>
</tr>
<tr>
<td>( \delta )</td>
<td>( \delta )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SVD</th>
<th>Exact</th>
<th>Uniform Leverage</th>
<th>Ridge leverage</th>
<th>Count-sketch</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORL</td>
<td>1.335</td>
<td>0.232</td>
<td>0.101</td>
<td>0.101</td>
<td>0.103</td>
</tr>
<tr>
<td>PEMS</td>
<td>35.781</td>
<td>3.770</td>
<td>0.917</td>
<td>0.892</td>
<td>0.970</td>
</tr>
</tbody>
</table>

Table 2: CPU wall-clock times (in seconds) for RFDA on ORL and PEMS.