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 compared to the conventional approach. Our analysis builds upon two simple structural results that boil down to randomized matrix multiplication, a fundamental and well-understood primitive of randomized linear algebra. We analyze the behavior of RFDA when leverage scores and ridge leverage scores are used to select predictor variables, and prove that accurate approximations can be achieved by a sample whose size depends on the effective degrees of freedom of the RFDA problem. Our results yield significant improvements over existing 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*

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

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

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

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

$$\mathbf{\Sigma}_b\, \mathbf{x}_i = \lambda_i\, \mathbf{\Sigma}_t\, \mathbf{x}_i, \quad i = 1, 2, \ldots q,$$

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

$$\mathbf{\Sigma}_b \mathbf{X} = \mathbf{\Sigma}_t \mathbf{X} \mathbf{\Lambda}, \tag{1}$$

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

$$(\mathbf{A}^\mathsf{T}\mathbf{\Omega}\mathbf{\Omega}^\mathsf{T}\mathbf{A})\mathbf{X} = (\mathbf{A}^\mathsf{T}\mathbf{A})\,\mathbf{X}\mathbf{\Lambda}, \tag{2}$$

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

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

If $\mathbf{A}^\mathsf{T}\mathbf{A}$ is non-singular, then $(\lambda_i, \mathbf{x}_i)$, $i = 1, 2, \ldots, q$ are the eigen-pairs of the matrix $(\mathbf{A}^\mathsf{T}\mathbf{A})^{-1}\mathbf{A}^\mathsf{T}\mathbf{\Omega}\mathbf{\Omega}^\mathsf{T}\mathbf{A}$. However, in many applications such as micro-array analysis [19], information retrieval [10], and face recognition [16, 37], the underlying $\mathbf{A}^\mathsf{T}\mathbf{A}$ is ill-conditioned as the number of predictors greatly exceeds the number of observations, *i.e.*, $d \gg n$. This makes the computation of $(\mathbf{A}^\mathsf{T}\mathbf{A})^{-1}\mathbf{A}^\mathsf{T}\mathbf{\Omega}\mathbf{\Omega}^\mathsf{T}\mathbf{A}$ numerically unstable. A popular alternative to FDA that addresses this problem is *regularized* Fisher discriminant analysis (RFDA) [17, 19].[2]

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

$$\mathbf{G}\,\mathbf{\Omega}^\mathsf{T}\mathbf{A}\mathbf{X} = \mathbf{X}\mathbf{\Lambda}, \tag{3}$$

where

$$\mathbf{G} = (\mathbf{A}^\mathsf{T}\mathbf{A} + \lambda\mathbf{I}_d)^{-1}\mathbf{A}^\mathsf{T}\mathbf{\Omega} = \mathbf{A}^\mathsf{T}(\mathbf{A}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)^{-1}\mathbf{\Omega}.$$

(The last equality can be verified using the SVD of $\mathbf{A}$.) Note that $\mathbf{A}^\mathsf{T}\mathbf{A} + \lambda\mathbf{I}_d$ is always invertible for $\lambda > 0$. We define the *effective degrees of freedom* of RFDA as

$$d_\lambda = \sum_{i=1}^{\rho} \frac{\sigma_i^2}{\sigma_i^2 + \lambda} \leq \rho\,. \tag{4}$$

Here, ρ is the rank of the matrix $\mathbf{A}$ and we note that d_λ depends on both the value of the regularization parameter λ and the non-zero singular values σ_i^2, $i = 1, 2, \ldots, \rho$.

Solving the RFDA problem of eqn. (3). Notice that the solution $(\mathbf{X}, \mathbf{\Lambda})$ to eqn. (3) may not be unique. Indeed, if $\mathbf{X}$ is a solution to eqn. (3), then for any non-singular diagonal matrix $\mathbf{D} \in \mathbb{R}^{q\times q}$, $\mathbf{X}\mathbf{D}$ is also a solution. [36] proposed an eigenvalue decomposition (EVD)-based algorithm (see Algorithm 2 in Appendix B) which not only returns $\mathbf{X}$ as a solution to eqn. (3) but also guarantees that for any two data points $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^d$, $\mathbf{X}$ satisfies $\|(\mathbf{w}_1 - \mathbf{w}_2)^\mathsf{T}\mathbf{X}\|_2 = \|(\mathbf{w}_1 - \mathbf{w}_2)^\mathsf{T}\mathbf{G}\|_2$ (see Theorem 8). This implies that instead of using the actual solution $\mathbf{X}$, if we project the points using $\mathbf{G}$, the distances between the projected points would also be preserved. Thus, for any distance-based classification method (e.g., k-nearest-neighbors), both $\mathbf{X}$ and $\mathbf{G}$ would result in the same predictions. Therefore, when solving eqn. (3) it is reasonable to shift our interest from $\mathbf{X}$ to $\mathbf{G}$. However, due to the high dimensionality d of the input data, exact computation of $\mathbf{G}$ is expensive, taking time $\mathcal{O}(n^2d + n^3 + ndc)$.

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 $\mathbf{S} \in \mathbb{R}^{d\times s}$ (for an appropriate choice of the sketching dimension $s \ll d$), such that

$$\|\mathbf{\Sigma}_\lambda\mathbf{V}^\mathsf{T}\mathbf{S}\mathbf{S}^\mathsf{T}\mathbf{V}\mathbf{\Sigma}_\lambda - \mathbf{\Sigma}_\lambda^2\|_2 \leq \frac{\varepsilon}{2}\,. \tag{5}$$

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

$$(\mathbf{\Sigma}_\lambda)_{ii} = \frac{\sigma_i}{\sqrt{\sigma_i^2 + \lambda}}, \qquad i = 1, \ldots, \rho\,. \tag{6}$$

Notice that $\|\mathbf{\Sigma}_\lambda\|_F^2 = d_\lambda$, 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 $\mathbf{S}$ with sketch-size s depending on d_λ (see Appendix F for details). Recall that d_λ is upper bounded by ρ but could be significantly smaller depending on the distribution of the singular values and the choice of λ. Indeed, it follows that by sampling-and-rescaling $\mathcal{O}(d_\lambda \ln d_\lambda)$ predictor variables from the matrix $\mathbf{A}$ (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 $\widehat{\mathbf{G}}$ satisfying

$$\|(\mathbf{w} - \mathbf{m})^\mathsf{T}(\widehat{\mathbf{G}} - \mathbf{G})\|_2 \leq \frac{\varepsilon^t}{\sqrt{\lambda}}\,\|\mathbf{V}\mathbf{V}^\mathsf{T}(\mathbf{w} - \mathbf{m})\|_2\,. \tag{7}$$

Here, $\mathbf{w} \in \mathbb{R}^d$ is any test data point and $\mathbf{V}\mathbf{V}^\mathsf{T}(\mathbf{w} - \mathbf{m})$ is the part of $\mathbf{w} - \mathbf{m}$ that lies within the range of $\mathbf{A}^\mathsf{T}$ (see footnote 1 for the definition of $\mathbf{m}$). We note that the dependency of the error on ε drops *exponentially* fast as the number of iterations t increases. See Section 2.2 for constructions of $\mathbf{S}$ 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:

$$\|\mathbf{V}^\mathsf{T}\mathbf{S}\mathbf{S}^\mathsf{T}\mathbf{V} - \mathbf{I}_\rho\|_2 \leq \frac{\varepsilon}{2}\,. \tag{8}$$

Indeed, assuming that the rank of $\mathbf{A}$ is much smaller than $\min\{n, d\}$, one can use the (exact or approximate) column *leverage scores* [22, 21] of the matrix $\mathbf{A}$ (*cf.* Section 1.3) to satisfy the aforementioned constraint by sampling $\mathcal{O}(\rho \ln \rho)$ columns, in which case $\mathbf{S}$ is a sampling-and-rescaling matrix. Perhaps more interestingly, a variety of oblivious sketching matrix constructions for $\mathbf{S}$ can also be used to satisfy eqn. (8) (see Section 2.2 for

[2] We note that another variant is *pseudo-inverse* FDA [29], which replaces $(\mathbf{A}^\mathsf{T}\mathbf{A})^{-1}$ by $(\mathbf{A}^\mathsf{T}\mathbf{A})^\dagger$.

specific constructions of $\mathbf{S}$). In either case, under this structural condition, the output of Algorithm 1 satisfies

$$\|(\mathbf{w}-\mathbf{m})^{\mathsf{T}}(\widehat{\mathbf{G}}-\mathbf{G})\|_2 \leq \frac{\varepsilon^t}{2\sqrt{\lambda}}\,\|\mathbf{V}\mathbf{V}^{\mathsf{T}}(\mathbf{w}-\mathbf{m})\|_2. \quad (9)$$

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 $\mathcal{O}(\rho \ln \rho)$ predictor variables from the matrix $\mathbf{A}$, 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 $\mathbf{A}$. More precisely, they proved that their method computes a matrix $\widehat{\mathbf{G}}$ in time $\mathcal{O}(\mathrm{nnz}(\mathbf{A})) + \mathcal{O}(n^2 s + n^3 + ndc)$,[3] which, for any test data point $\mathbf{w} \in \mathbb{R}^d$, satisfies

$$\|(\mathbf{w}-\mathbf{m})^{\mathsf{T}}(\widehat{\mathbf{G}}-\mathbf{G})\|_2 \leq \frac{\kappa\,\varepsilon}{1-\varepsilon}\|\mathbf{V}\mathbf{V}^{\mathsf{T}}(\mathbf{w}-\mathbf{m})\|_2\,,$$

with high probability for any $\varepsilon \in (0, 1]$ (here, κ is the condition number of $\mathbf{A}$). Thus, their random projection–based approach well-approximates the original RFDA problem only when $\mathbf{A}$ is well-conditioned (*i.e.*, κ small). In addition, the running time of their approach grows proportionally to $\mathcal{O}(1/\varepsilon^2)$, whereas our algorithm runs in $\mathcal{O}(\log(1/\varepsilon))$ time (*cf.* Section 2.2). Lastly, our main result depends only on the effective degrees of freedom d_λ (*cf.* Theorem 1), which can be much smaller than ρ.

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 $\mathbf{G}$ has similar properties with the solution matrix $\mathbf{X}$ 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 $\mathbf{A}$ 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 λ and the singular values of $\mathbf{A}$, 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 RFDA 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, $s = \mathcal{O}(\rho^2/\delta\varepsilon^2)$, where δ is the failure probability.

the actual FDA stage. Another line of research [24, 31] dealt with the fast implementation of 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* $\mathbf{\Sigma}_t - \mathbf{\Sigma}_b$, 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 $\mathbf{A}^\mathsf{T}\mathbf{A}$, 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 $\mathbf{a}, \mathbf{b}, \ldots$ to denote vectors and $\mathbf{A}, \mathbf{B}, \ldots$ to denote matrices. For a matrix $\mathbf{A}$, $\mathbf{A}_{*i}$ ($\mathbf{A}_{i*}$) denotes the i-th column (row) of $\mathbf{A}$ as a column (row) vector. For a vector $\mathbf{a}$, $\|\mathbf{a}\|_2$ denotes its Euclidean norm; for a matrix $\mathbf{A}$, $\|\mathbf{A}\|_2$ denotes its spectral norm and $\|\mathbf{A}\|_F$ 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 $\mathbf{A} \in \mathbb{R}^{n\times d}$ with $d \geq n$ of rank ρ, its (*thin*) Singular Value Decomposition (SVD) is the product $\mathbf{U\Sigma V}^\mathsf{T}$, with $\mathbf{U} \in \mathbb{R}^{n\times\rho}$ (the matrix of the left singular vectors), $\mathbf{V} \in \mathbb{R}^{d\times\rho}$ (the matrix of the right singular vectors), and $\mathbf{\Sigma} \in \mathbb{R}^{\rho\times\rho}$ a diagonal matrix whose diagonal entries are the non-zero singular values of $\mathbf{A}$ arranged in non-increasing order. Computation of the SVD takes, in this setting, $\mathcal{O}(n^2 d)$ time. We will often use σ_i to denote the singular values of a matrix implied by context.

We shall also make use of the *full* SVD representation $\mathbf{A} = \mathbf{U}_f\mathbf{\Sigma}_f\mathbf{V}_f^\mathsf{T}$, where $\mathbf{U}_f = (\mathbf{U}\ \ \mathbf{U}_\perp) \in \mathbb{R}^{n\times n}$, $\mathbf{V}_f = (\mathbf{V}\ \ \mathbf{V}_\perp) \in \mathbb{R}^{d\times d}$, and $\mathbf{\Sigma}_f = \begin{pmatrix} \mathbf{\Sigma}_{\rho\times\rho} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{n\times d}$. Here, $\mathbf{U}_\perp \in \mathbb{R}^{n\times(n-\rho)}$ and $\mathbf{V}_\perp \in \mathbb{R}^{d\times(d-\rho)}$. Finally, the column *leverage scores* and *ridge leverage scores* of $\mathbf{A}$ are respectively given by $\|\mathbf{V}_{i*}\|_2^2$ and $\|(\mathbf{V\Sigma}_\lambda)_{i*}\|_2^2$ for $i = 1, 2, \ldots, d$. (Recall the definition of $\mathbf{\Sigma}_\lambda$ 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 while 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 $\widetilde{\mathbf{G}}^{(j)} \in \mathbb{R}^{d\times c}$ for $j = 1, \ldots, t$ and returns the estimator $\widehat{\mathbf{G}} = \sum_{j=1}^{t} \widetilde{\mathbf{G}}^{(j)}$ to the original matrix $\mathbf{G}$ 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: $\mathbf{A} \in \mathbb{R}^{n\times d}$, $\mathbf{\Omega} \in \mathbb{R}^{n\times c}$, $\lambda > 0$; number of iterations $t > 0$; sketching matrix $\mathbf{S} \in \mathbb{R}^{d\times s}$;
Initialize: $\mathbf{L}^{(0)} \leftarrow \mathbf{\Omega}$, $\widetilde{\mathbf{G}}^{(0)} \leftarrow \mathbf{0}_{d\times c}$, $\mathbf{Y}^{(0)} \leftarrow \mathbf{0}_{n\times c}$;
for $j = 1$ **to** t **do**
  $\mathbf{L}^{(j)} \leftarrow \mathbf{L}^{(j-1)} - \lambda\mathbf{Y}^{(j-1)} - \mathbf{A}\widetilde{\mathbf{G}}^{(j-1)}$;
  $\mathbf{Y}^{(j)} \leftarrow (\mathbf{ASS}^\mathsf{T}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)^{-1}\mathbf{L}^{(j)}$;
  $\widetilde{\mathbf{G}}^{(j)} \leftarrow \mathbf{A}^\mathsf{T}\mathbf{Y}^{(j)}$;
end for
Output: $\widehat{\mathbf{G}} = \sum_{j=1}^{t} \widetilde{\mathbf{G}}^{(j)}$;

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

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

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

Recall that $\mathbf{VV}^\mathsf{T}(\mathbf{w}-\mathbf{m})$ is the projection of the vector $\mathbf{w}-\mathbf{m}$ onto the row space of $\mathbf{A}$.

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

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

$$\|(\mathbf{w}-\mathbf{m})^\mathsf{T}(\widehat{\mathbf{G}} - \mathbf{G})\|_2 \leq \frac{\varepsilon^t}{2\sqrt{\lambda}} \|\mathbf{VV}^\mathsf{T}(\mathbf{w}-\mathbf{m})\|_2 .$$

Recall that $\mathbf{VV}^\mathsf{T}(\mathbf{w}-\mathbf{m})$ is the projection of the vector $\mathbf{w}-\mathbf{m}$ onto the row space of $\mathbf{A}$.

Running time of Algorithm 1. First, we need to compute $\mathbf{A}\widetilde{\mathbf{G}}^{(j-1)}$ which takes time $\mathcal{O}(c \cdot \mathrm{nnz}(\mathbf{A}))$. Then, computing the sketch $\mathbf{AS} \in \mathbb{R}^{n\times s}$ takes $T(\mathbf{A}, \mathbf{S})$ time

which depends on the particular construction of $\mathbf{S}$ (see Section 2.2). In order to invert the matrix $\mathbf{\Theta} = \mathbf{A}\mathbf{S}\mathbf{S}^\mathsf{T}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n$, it suffices to compute the SVD of the matrix $\mathbf{AS}$. Notice that given the singular values of $\mathbf{AS}$ we can compute the singular values of $\mathbf{\Theta}$ and also notice that the left and right singular vectors of $\mathbf{\Theta}$ are the same as the left singular vectors of $\mathbf{AS}$. Interestingly, we do not need to compute $\mathbf{\Theta}^{-1}$: we can store it implicitly by storing its left (and right) singular vectors $\mathbf{U}_{\mathbf{\Theta}}$ and its singular values $\mathbf{\Sigma}_{\mathbf{\Theta}}$. Then, we can compute all necessary matrix-vector products using this implicit representation of $\mathbf{\Theta}^{-1}$. Thus, inverting $\mathbf{\Theta}$ takes $\mathcal{O}(sn^2)$ time. Updating the matrices $\mathbf{L}^{(j)}$, $\mathbf{Y}^{(j)}$, and $\widetilde{\mathbf{G}}^{(j)}$ is dominated by the aforementioned running times. Thus, summing over all t iterations, the running time of Algorithm 1 is

$$\mathcal{O}(t\,c\cdot\mathrm{nnz}(\mathbf{A})) + \mathcal{O}(s\,n^2) + T(\mathbf{A},\mathbf{S}), \tag{10}$$

which should be compared to the $\mathcal{O}(n^2 d)$ 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 $(\mathbf{A}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)\mathbf{F} = \mathbf{\Omega}$ in $\mathbf{F} \in \mathbb{R}^{n\times c}$ with randomized pre-conditioner $\mathbf{P}^{-1} = (\mathbf{A}\mathbf{S}\mathbf{S}^\mathsf{T}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)^{-1}$. However, our objective and analysis are significantly different compared to the conventional preconditioned Richardson iteration. *First*, our matrix of interest is $\mathbf{G} = \mathbf{A}^\mathsf{T}\mathbf{F} \in \mathbb{R}^{d\times c}$, whereas standard analysis of the preconditioned Richardson iteration is with respect to $\mathbf{F}$. Specifically, in the context of discriminant analysis, for a new observation $\mathbf{w} \in \mathbb{R}^d$, we are interested in understanding whether the output of our algorithm closely approximates the original point in the projected space, *i.e.*, if $\|(\mathbf{w}-\mathbf{m})^\mathsf{T}(\widehat{\mathbf{G}} - \mathbf{G})\|_2$ is sufficiently small. To the best of our knowledge, standard analysis of preconditioned Richardson iteration does not yield a bound for $\|(\mathbf{w}-\mathbf{m})^\mathsf{T}(\widehat{\mathbf{G}} - \mathbf{G})\|_2$. *Second*, our analysis is with respect to the Euclidean norm whereas the standard analysis is in terms of the energy-norm of $(\mathbf{A}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)$, as the matrix $\mathbf{P}^{-1}(\mathbf{A}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)$ is not symmetric positive definite.

Finally, we note that our proof also holds when *different* sampling matrices $\mathbf{S}_j$ (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 $\mathbf{S}_j$ do not even need to have the same number of columns (see Section 5 for an open problem in this setting).

2.2 SATISFYING THE STRUCTURAL CONDITIONS

The structural conditions of eqns. (5) and (8) essentially boil down to randomized, approximate matrix multiplication [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 [9]. Under our notation, [9] proved that for $\mathbf{Z} \in \mathbb{R}^{d\times n}$ and for a (suitably constructed) sketching matrix $\mathbf{S} \in \mathbb{R}^{d\times s}$, with probability at least $1-\delta$,

$$\|\mathbf{Z}^\mathsf{T}\mathbf{S}\mathbf{S}^\mathsf{T}\mathbf{Z} - \mathbf{Z}^\mathsf{T}\mathbf{Z}\|_2 \le \varepsilon\left(\|\mathbf{Z}\|_2^2 + \frac{\|\mathbf{Z}\|_F^2}{r}\right). \tag{11}$$

This bound holds for a broad family of constructions for the sketching matrix $\mathbf{S}$ (see [9] for details). In particular, [9] demonstrated a construction for $\mathbf{S}$ with $s = \mathcal{O}(r/\varepsilon^2)$ columns such that, for any $n\times d$ matrix $\mathbf{A}$, the product $\mathbf{AS}$ can be computed in time $\mathcal{O}(\mathrm{nnz}(\mathbf{A})) + \tilde{\mathcal{O}}((r^3 + r^2 n)/\varepsilon^\gamma)$ for some constant γ. Thus, starting with eqn. (5) and using this particular construction for $\mathbf{S}$, let $\mathbf{Z} = \mathbf{V}\mathbf{\Sigma}_\lambda$ and note that $\|\mathbf{V}\mathbf{\Sigma}_\lambda\|_F^2 = d_\lambda$ and $\|\mathbf{V}\mathbf{\Sigma}_\lambda\|_2 \le 1$. Setting $r = d_\lambda$, eqn. (11) implies that

$$\|\mathbf{\Sigma}_\lambda\mathbf{V}^\mathsf{T}\mathbf{S}\mathbf{S}^\mathsf{T}\mathbf{V}\mathbf{\Sigma}_\lambda - \mathbf{\Sigma}_\lambda^2\|_2 \le 2\,\varepsilon.$$

In this case, the running time needed to compute the sketch equals $T(\mathbf{A},\mathbf{S}) = \mathcal{O}(\mathrm{nnz}(\mathbf{A})) + \widetilde{\mathcal{O}}(d_\lambda^2 n/\varepsilon^\gamma)$. The running time of the overall algorithm follows from eqn. (10) and our choices for s and r:

$$\mathcal{O}(t\,c\cdot\mathrm{nnz}(\mathbf{A})) + \widetilde{\mathcal{O}}(d_\lambda n^2/\varepsilon^{\max\{2,\gamma\}}).$$

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

Similarly, starting with eqn. (8), let $\mathbf{Z} = \mathbf{V}$ and note that $\|\mathbf{V}\|_F^2 = \rho$ and $\|\mathbf{V}\|_2 = 1$. Setting $r = \rho$, eqn. (11) implies that $\|\mathbf{V}^\mathsf{T}\mathbf{S}\mathbf{S}^\mathsf{T}\mathbf{V} - \mathbf{I}_\rho\|_2 \le 2\varepsilon$. In this case, the running time of the sketch computation is equal to $T(\mathbf{A},\mathbf{S}) = \mathcal{O}(\mathrm{nnz}(\mathbf{A})) + \widetilde{\mathcal{O}}(\rho^2 n/\varepsilon^\gamma)$. The running time of the overall algorithm follows from eqn. (10) and our choices for s and r:

$$\mathcal{O}(t\,c\cdot\mathrm{nnz}(\mathbf{A})) + \widetilde{\mathcal{O}}(\rho n^2/\varepsilon^{\max\{2,\gamma\}}).$$

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_λ) of the problem (or, on the rank ρ of the data matrix $\mathbf{A}$). In this case, the SVD of $\mathbf{AS}$ can be computed in $\mathcal{O}(ns^2)$ time, and the running time of our algorithm is given by $\mathcal{O}(t\,c\cdot\mathrm{nnz}(\mathbf{A})) + \widetilde{\mathcal{O}}(d_\lambda^2 n/\varepsilon^{\max\{4,\gamma\}})$ (or, $\mathcal{O}(t\,c\cdot\mathrm{nnz}(\mathbf{A})) + \widetilde{\mathcal{O}}(\rho^2 n/\varepsilon^{\max\{4,\gamma\}})$).

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

$$\mathbf{G}^{(j)} = \mathbf{A}^\mathsf{T}(\mathbf{A}\mathbf{A}^\mathsf{T} + \lambda \mathbf{I}_n)^{-1}\mathbf{L}^{(j)}, \quad j = 1, \ldots, t. \tag{12}$$

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

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

$$\mathbf{G}^{(j)} = \mathbf{V}\mathbf{\Sigma}_\lambda^2\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j)}. \tag{13}$$

Proof. Using the full SVD representation of $\mathbf{A}$, we have

$$\begin{aligned}
\mathbf{G}^{(j)} &= \mathbf{V}_f\mathbf{\Sigma}_f^\mathsf{T}\mathbf{U}_f^\mathsf{T}(\mathbf{U}_f\mathbf{\Sigma}_f\mathbf{\Sigma}_f^\mathsf{T}\mathbf{U}_f^\mathsf{T} + \lambda\mathbf{U}_f\mathbf{U}_f^\mathsf{T})^{-1}\mathbf{L}^{(j)} \\
&= \mathbf{V}_f\mathbf{\Sigma}_f^\mathsf{T}(\mathbf{\Sigma}_f\mathbf{\Sigma}_f^\mathsf{T} + \lambda\mathbf{I}_n)^{-1}\mathbf{U}_f^\mathsf{T}\mathbf{L}^{(j)} \\
&= \begin{pmatrix}\mathbf{V} & \mathbf{V}_\perp\end{pmatrix}\begin{pmatrix}\mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{pmatrix}\left[\begin{pmatrix}\mathbf{\Sigma}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{pmatrix} + \lambda\mathbf{I}_n\right]^{-1}\begin{pmatrix}\mathbf{U}^\mathsf{T} \\ \mathbf{U}_\perp^\mathsf{T}\end{pmatrix}\mathbf{L}^{(j)} \\
&= \begin{pmatrix}\mathbf{V} & \mathbf{V}_\perp\end{pmatrix}\begin{pmatrix}\mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{pmatrix}\left[\begin{pmatrix}\mathbf{\Sigma}^2 + \lambda\mathbf{I}_\rho & \mathbf{0} \\ \mathbf{0} & \lambda\mathbf{I}_{n-\rho}\end{pmatrix}\right]^{-1}\begin{pmatrix}\mathbf{U}^\mathsf{T} \\ \mathbf{U}_\perp^\mathsf{T}\end{pmatrix}\mathbf{L}^{(j)} \\
&= \begin{pmatrix}\mathbf{V} & \mathbf{V}_\perp\end{pmatrix}\begin{pmatrix}\mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{pmatrix}\begin{pmatrix}(\mathbf{\Sigma}^2 + \lambda\mathbf{I}_\rho)^{-1} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\lambda}\mathbf{I}_{n-\rho}\end{pmatrix}\begin{pmatrix}\mathbf{U}^\mathsf{T} \\ \mathbf{U}_\perp^\mathsf{T}\end{pmatrix}\mathbf{L}^{(j)} \\
&= \begin{pmatrix}\mathbf{V} & \mathbf{V}_\perp\end{pmatrix}\begin{pmatrix}\mathbf{\Sigma}(\mathbf{\Sigma}^2 + \lambda\mathbf{I}_\rho)^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{U}^\mathsf{T} \\ \mathbf{U}_\perp^\mathsf{T}\end{pmatrix}\mathbf{L}^{(j)} \\
&= \mathbf{V}\mathbf{\Sigma}(\mathbf{\Sigma}^2 + \lambda\mathbf{I}_\rho)^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j)} \\
&= \mathbf{V}\mathbf{\Sigma}\mathbf{\Sigma}^{-1}(\mathbf{I}_\rho + \lambda\mathbf{\Sigma}^{-2})^{-1}\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j)} \\
&= \mathbf{V}\mathbf{\Sigma}_\lambda^2\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j)},
\end{aligned}$$

which completes the proof. □

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 $\mathbf{L}^{(j)}$ be as defined in Algorithm 1 and let $\widetilde{\mathbf{G}}^{(j)}$ be defined as in eqn.* (12)*. Further, let $\mathbf{S} \in \mathbb{R}^{d\times s}$ be the sketching matrix and let $\mathbf{E} = \mathbf{\Sigma}_\lambda\mathbf{V}^\mathsf{T}\mathbf{S}\mathbf{S}^\mathsf{T}\mathbf{V}\mathbf{\Sigma}_\lambda - \mathbf{\Sigma}_\lambda^2$. If eqn.* (5) *is satisfied, i.e., $\|\mathbf{E}\|_2 \leq \frac{\varepsilon}{2}$, then, for all $j = 1, \ldots, t$,*

$$\begin{aligned}
&\|(\mathbf{w} - \mathbf{m})^\mathsf{T}(\widetilde{\mathbf{G}}^{(j)} - \mathbf{G}^{(j)})\|_2 \\
&\quad \leq \varepsilon\,\|\mathbf{V}\mathbf{V}^\mathsf{T}(\mathbf{w} - \mathbf{m})\|_2\,\|\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j)}\|_2\,.
\end{aligned} \tag{14}$$

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

$$\widetilde{\mathbf{G}}^{(j)} = \mathbf{G}^{(j)} + \mathbf{V}\mathbf{\Sigma}_\lambda\mathbf{Q}\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j)}, \tag{15}$$

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

$$\begin{aligned}
\|\mathbf{Q}\|_2 &= \Big\|\sum_{\ell=1}^{\infty}(-1)^\ell\mathbf{E}^\ell\Big\|_2 \leq \sum_{\ell=1}^{\infty}\|\mathbf{E}^\ell\|_2 \\
&\leq \sum_{\ell=1}^{\infty}\|\mathbf{E}\|_2^\ell \leq \sum_{\ell=1}^{\infty}\left(\frac{\varepsilon}{2}\right)^\ell = \frac{\varepsilon/2}{1 - \varepsilon/2} \leq \varepsilon\,.
\end{aligned} \tag{16}$$

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

$$\begin{aligned}
&\|(\mathbf{w} - \mathbf{m})^\mathsf{T}(\widetilde{\mathbf{G}}^{(j)} - \mathbf{G}^{(j)})\|_2 \\
&\quad = \|(\mathbf{w} - \mathbf{m})^\mathsf{T}\mathbf{V}\mathbf{\Sigma}_\lambda\mathbf{Q}\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j)}\|_2 \\
&\quad \leq \|(\mathbf{w} - \mathbf{m})^\mathsf{T}\mathbf{V}\|_2\,\|\mathbf{\Sigma}_\lambda\|_2\,\|\mathbf{Q}\|_2\,\|\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j)}\|_2 \\
&\quad \leq \varepsilon\,\|\mathbf{V}\mathbf{V}^\mathsf{T}(\mathbf{w} - \mathbf{m})\|_2\,\|\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j)}\|_2,
\end{aligned}$$

where the last inequality follows from eqn. (16) and the fact that $\|\mathbf{\Sigma}_\lambda\|_2 \leq 1$. □

The next lemma presents a structural result for $\mathbf{G}$.

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

$$\mathbf{G} = \mathbf{G}^{(t)} + \sum_{j=1}^{t-1}\widetilde{\mathbf{G}}^{(j)}. \tag{17}$$

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

$$\mathbf{G}^{(1)} = \mathbf{A}^\mathsf{T}(\mathbf{A}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)^{-1}\mathbf{L}^{(1)} = \mathbf{G}.$$

For $t = 2$, we get

$$\begin{aligned}
\mathbf{G}^{(2)} &= \mathbf{A}^\mathsf{T}(\mathbf{A}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)^{-1}\mathbf{L}^{(2)} \\
&= \mathbf{A}^\mathsf{T}(\mathbf{A}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)^{-1}(\mathbf{L}^{(1)} - \lambda\mathbf{Y}^{(1)} - \mathbf{A}\widetilde{\mathbf{G}}^{(1)}) \\
&= \mathbf{A}^\mathsf{T}(\mathbf{A}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)^{-1}\mathbf{L}^{(1)}
\end{aligned}$$

$$- \mathbf{A}^\mathsf{T}(\mathbf{A}\mathbf{S}\mathbf{S}^\mathsf{T}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)^{-1}\mathbf{L}^{(1)} \quad (18)$$
$$= \mathbf{G} - \widetilde{\mathbf{G}}^{(1)}.$$

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

$$\mathbf{G}^{(p)} = \mathbf{G} - \sum_{j=1}^{p-1} \widetilde{\mathbf{G}}^{(j)}. \quad (19)$$

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

$$\begin{aligned}
\mathbf{G}^{(p+1)} &= \mathbf{A}^\mathsf{T}(\mathbf{A}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)^{-1}\mathbf{L}^{(p+1)} \\
&= \mathbf{A}^\mathsf{T}(\mathbf{A}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)^{-1}(\mathbf{L}^{(p)} - \lambda\mathbf{Y}^{(p)} - \mathbf{A}\widetilde{\mathbf{G}}^{(p)}) \\
&= \mathbf{A}^\mathsf{T}(\mathbf{A}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)^{-1}\mathbf{L}^{(p)} \\
&\qquad - \mathbf{A}^\mathsf{T}(\mathbf{A}\mathbf{S}\mathbf{S}^\mathsf{T}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)^{-1}\mathbf{L}^{(p)} \quad (20) \\
&= \mathbf{G}^{(p)} - \widetilde{\mathbf{G}}^{(p)} = \left(\mathbf{G} - \textstyle\sum_{j=1}^{p-1} \widetilde{\mathbf{G}}^{(j)}\right) - \widetilde{\mathbf{G}}^{(p)} \\
&= \mathbf{G} - \textstyle\sum_{j=1}^{p} \widetilde{\mathbf{G}}^{(j)},
\end{aligned}$$

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

Repeated application of Lemmas 5 and 4 yields:

$$\begin{aligned}
&\|(\mathbf{w} - \mathbf{m})^\mathsf{T}(\widehat{\mathbf{G}} - \mathbf{G})\|_2 \\
&= \|(\mathbf{w} - \mathbf{m})^\mathsf{T}(\textstyle\sum_{j=1}^{t} \widetilde{\mathbf{G}}^{(j)} - \mathbf{G})\|_2 \quad (21) \\
&= \|(\mathbf{w} - \mathbf{m})^\mathsf{T}(\widetilde{\mathbf{G}}^{(t)} - (\mathbf{G} - \textstyle\sum_{j=1}^{t-1} \widetilde{\mathbf{G}}^{(j)}))\|_2 \\
&\leq \|(\mathbf{w} - \mathbf{m})^\mathsf{T}(\widetilde{\mathbf{G}}^{(t)} - \mathbf{G}^{(t)})\|_2 \\
&\leq \varepsilon \, \|\mathbf{V}\mathbf{V}^\mathsf{T}(\mathbf{w} - \mathbf{m})\|_2 \|\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(t)}\|_2. \quad (22)
\end{aligned}$$

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., $\|\mathbf{E}\|_2 \leq \frac{\varepsilon}{2}$, then*

$$\|\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j+1)}\|_2 \leq \varepsilon \, \|\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j)}\|_2. \quad (23)$$

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

$$\begin{aligned}
\mathbf{L}^{(j+1)} &= \mathbf{L}^{(j)} - \lambda\mathbf{Y}^{(j)} - \mathbf{A}\widetilde{\mathbf{G}}^{(j)} \\
&= \mathbf{L}^{(j)} - (\mathbf{A}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)(\mathbf{A}\mathbf{S}\mathbf{S}^\mathsf{T}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)^{-1}\mathbf{L}^{(j)}. \quad (24)
\end{aligned}$$

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

$$\begin{aligned}
&(\mathbf{A}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)(\mathbf{A}\mathbf{S}\mathbf{S}^\mathsf{T}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)^{-1}\mathbf{L}^{(j)} \\
&= \mathbf{L}^{(j)} + \mathbf{U}(\mathbf{\Sigma}^2 + \lambda\mathbf{I}_\rho)\mathbf{\Sigma}^{-1}\mathbf{\Sigma}_\lambda\mathbf{Q}\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j)}, \quad (25)
\end{aligned}$$

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

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

$$\mathbf{L}^{(j+1)} = -\mathbf{U}(\mathbf{\Sigma}^2 + \lambda\mathbf{I}_\rho)\mathbf{\Sigma}^{-1}\mathbf{\Sigma}_\lambda\mathbf{Q}\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j)}. \quad (26)$$

Finally, applying eqn. (26), we obtain

$$\begin{aligned}
&\|\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j+1)}\|_2 \\
&= \|\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{U}(\mathbf{\Sigma}^2 + \lambda\mathbf{I}_\rho)\mathbf{\Sigma}^{-1}\mathbf{\Sigma}_\lambda\mathbf{Q}\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j)}\|_2 \\
&= \|\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}(\mathbf{\Sigma}^2 + \lambda\mathbf{I}_\rho)\mathbf{\Sigma}^{-1}\mathbf{\Sigma}_\lambda\mathbf{Q}\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j)}\|_2 \\
&= \|\mathbf{Q}\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j)}\|_2 \leq \|\mathbf{Q}\|_2\|\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j)}\|_2 \\
&\leq \varepsilon \, \|\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(j)}\|_2
\end{aligned}$$

where the third equality holds since $\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}(\mathbf{\Sigma}^2 + \lambda\mathbf{I}_\rho)\mathbf{\Sigma}^{-1}\mathbf{\Sigma}_\lambda = \mathbf{I}_\rho$. The last two inequalities follow from sub-multiplicativity and the fact that $\|\mathbf{Q}\|_2 \leq \varepsilon$ (by eqn. (16)). □

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

$$\begin{aligned}
\|\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(t)}\|_2 &\leq \varepsilon \, \|\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(t-1)}\|_2 \\
&\leq \cdots \leq \varepsilon^{t-1}\|\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(1)}\|_2. \quad (27)
\end{aligned}$$

Notice that $\mathbf{L}^{(1)} = \mathbf{\Omega}$ by definition. Also, $\mathbf{\Omega}^\mathsf{T}\mathbf{\Omega} = \mathbf{I}_c$ and thus $\|\mathbf{\Omega}\|_2 = 1$. Furthermore, we know that $\|\mathbf{U}^\mathsf{T}\|_2 = 1$ and $\|\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\|_2 = \max_{1\leq i\leq\rho}(\sigma_i^2 + \lambda)^{-\frac{1}{2}}$. Thus, sub-multiplicativity yields

$$\begin{aligned}
\|\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\mathbf{U}^\mathsf{T}\mathbf{L}^{(1)}\|_2 &\leq \|\mathbf{\Sigma}_\lambda\mathbf{\Sigma}^{-1}\|_2\|\mathbf{U}^\mathsf{T}\|_2\|\mathbf{\Omega}\|_2 \\
&= \max_{1\leq i\leq\rho}(\sigma_i^2 + \lambda)^{-\frac{1}{2}} \leq \lambda^{-\frac{1}{2}}, \quad (28)
\end{aligned}$$

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

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

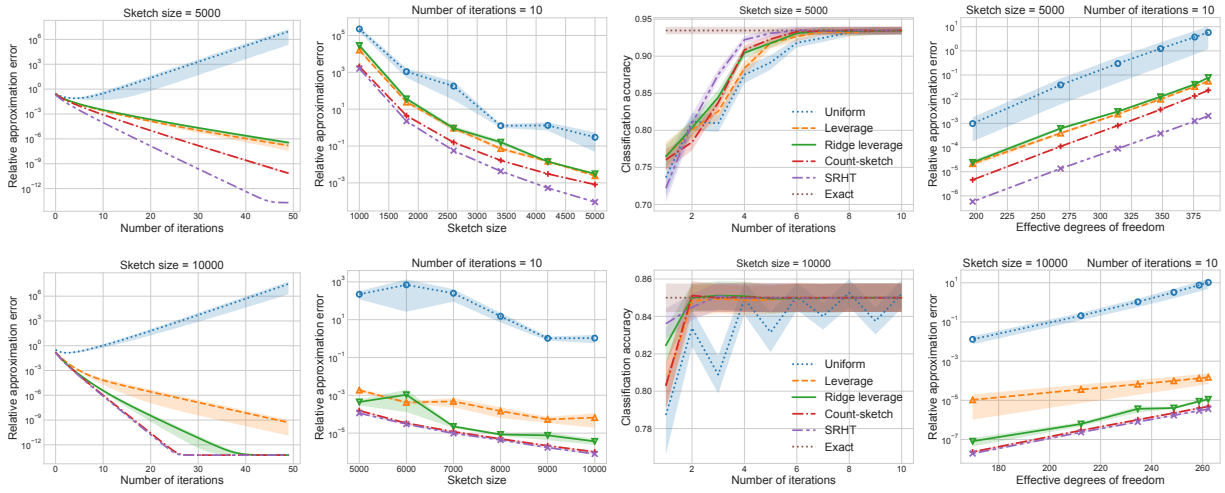
$$\|(\mathbf{w} - \mathbf{m})^\mathsf{T}(\widehat{\mathbf{G}} - \mathbf{G})\|_2 \leq \frac{\varepsilon^t}{\sqrt{\lambda}} \, \|\mathbf{V}\mathbf{V}^\mathsf{T}(\mathbf{w} - \mathbf{m})\|_2 \,,$$

which concludes the proof. □

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



(a) Error vs. iterations (b) Error vs. sketch size (c) Accuracy vs. iterations (d) Error vs. d_λ

Figure 1: Experiment results on ORL (top row) and PEMS (bottom row); errors are on log-scale.

$n = 400$ 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 $\mathbf{S}$. For sketching-based approaches (*cf.* Section 2.2), we construct $\mathbf{S}$ 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 $\mathbf{S}$ (*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 $\mathbf{S}$ with uniform sampling probabilities do not in general satisfy the structural conditions of eqns. (5) and (8).

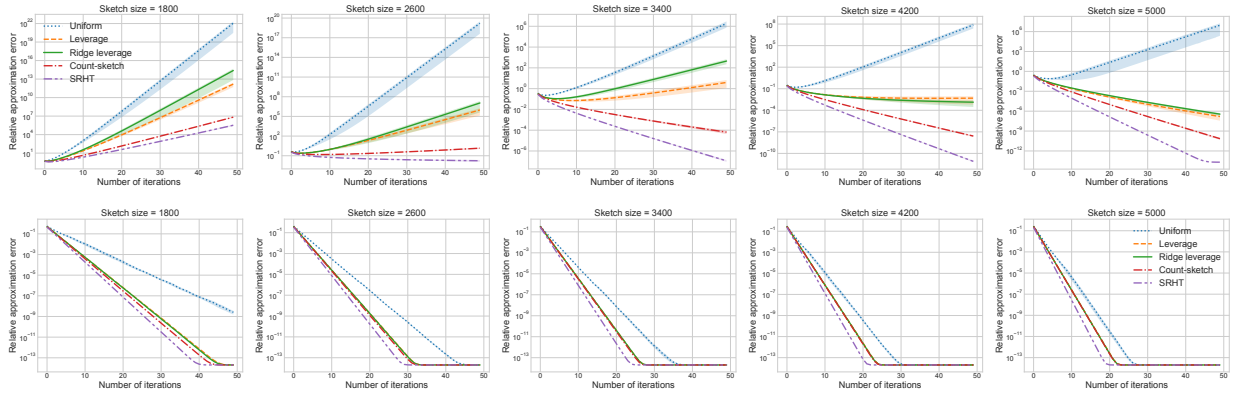
For each sketching method, we run Algorithm 1 for 50 iterations with a variety of sketch sizes, and measure the relative approximation error $\|\widehat{\mathbf{G}} - \mathbf{G}\|_F / \|\mathbf{G}\|_F$, where $\mathbf{G}$ is computed exactly. We also randomly divide each dataset into a training set with 60% examples and a test set of 40% examples (stratified by label), and measure the classification accuracy on the test set with $\widehat{\mathbf{G}}$ 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 $\widehat{\mathbf{G}} = \sum_{j=1}^{t} \widetilde{\mathbf{G}}^{(j)}$ 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 $\mathbf{S}$ 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 $s = \mathcal{O}(\varepsilon^{-2} d_\lambda \ln d_\lambda)$ suffices to satisfy the structural condition of eqn. (5), while for leverage scores, setting $s = \mathcal{O}(\varepsilon^{-2} \rho \ln \rho)$ suffices to satisfy the structural condition of eqn. (8). (Recall that ρ can be substantially larger than the effective degrees of freedom d_λ.) 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.



(a) $s = 1800$ (b) $s = 2600$ (c) $s = 3400$ (d) $s = 4200$ (e) $s = 5000$

Figure 2: Relative approximation error (on log-scale) vs. number of iterations on ORL dataset for increasing sketch size s. *Top row*: using a single sketching matrix $\mathbf{S}$ throughout. *Bottom row*: sample a new $\mathbf{S}_j$ at every iteration j.

In the last column of Figure 1, we keep the design matrix unchanged (fixing n) while varying the regularization parameter λ, and plot the relative approximation error against the effective degrees of freedom d_λ of the RFDA problem. We observe that the relative approximation error decreases exponentially as d_λ decreases; thus, the sketch size or number of iterations necessary to achieve a certain approximation precision also decreases with d_λ, 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

[1] N. Ailon and B. Chazelle. The fast Johnson–Lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, 39(1):302–322, 2009.

[2] A. E. Alaoui and M. W. Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, pages 775–783, 2015.

[3] AT&T Laboratories Cambridge. The ORL Database of Faces, 1994. Data retrieved from `http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`.

[4] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[5] L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33:1713–1726, 2000.

[6] A. Chowdhury, J. Yang, and P. Drineas. An iterative, sketching-based framework for ridge regression. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 988–997, 2018.

[7] K. L. Clarkson and D. P. Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, pages 81–90, 2013.

[8] M. B. Cohen, C. Musco, and C. Musco. Input sparsity time low-rank approximation via ridge leverage score sampling. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1758–1777, 2017.

[9] M. B. Cohen, J. Nelson, and D. P. Woodruff. Optimal approximate matrix product in terms of stable rank. In *43rd International Colloquium on Automata, Languages, and Programming*, pages 11:1–11:14, 2016.

[10] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[11] P. Drineas and R. Kannan. Fast monte-carlo algorithms for approximate matrix multiplication. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, pages 452–459, 2001.

[12] P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices I: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157, 2006.

[13] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13(1):3475–3506, 2012.

[14] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós. Faster least squares approximation. *Numerische Mathematik*, 117:219–249, 2011.

[15] R. J. Durrant and A. Kabán. A tight bound on the performance of Fisher's linear discriminant in randomly projected data spaces. *Pattern Recognition Letters*, 33(7):911–919, 2012.

[16] K. Etemad and R. Chellappa. Discriminant analysis for recognition of human face images. *Journal of the Optical Society of America A*, 14(8):1724–1733, 1997.

[17] J. H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989.

[18] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.

[19] Y. Guo, T. Hastie, and R. Tibshirani. Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, 8(1):86–100, 2007.

[20] J. T. Holodnak and I. C. F. Ipsen. Randomized approximation of the gram matrix: Exact computation and probabilistic bounds. *SIAM Journal on Matrix Analysis and Applications*, 36(1):110–137, 2015.

[21] M. W. Mahoney. *Randomized algorithms for matrices and data*. Foundations and Trends in Machine Learning. 2011.

[22] M. W. Mahoney and P. Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences of the United States of America*, 106(3), 2009.

[23] M. Pilanci and M. J. Wainwright. Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares. *Journal of Machine Learning Research*, 17(53):1–38, 2016.

[24] A. Sharma and K. K. Paliwal. A new perspective to null linear discriminant analysis method and its fast implementation using random matrix multiplication with scatter matrices. *Pattern Recognition*, 45(6):2205–2213, 2012.

[25] B. Tu, Z. Zhang, S. Wang, and H. Qian. Making Fisher discriminant analysis scalable. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 964–972, 2014.

[26] UCI Machine Learning Repository. PEMS-SF data set, 2011. Data retrieved from `https://archive.ics.uci.edu/ml/datasets/PEMS-SF`.

[27] J. Wang, J. Lee, M. Mahdavi, M. Kolar, and N. Srebro. Sketching meets random projection in the dual: A provable recovery algorithm for big and high-dimensional data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 1150–1158, 2017.

[28] S. Wang, A. Gittens, and M. W. Mahoney. Sketched ridge regression: Optimization perspective, statistical perspective, and model averaging. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3608–3616, 2017.

[29] A. R. Webb. *Linear Discriminant Analysis*. Wiley-Blackwell, 2003.

[30] D. P. Woodruff. Sketching as a Tool for Numerical Linear Algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1-2):1–157, 2014.

[31] G. Wu and T. Feng. A theoretical contribution to the fast implementation of null linear discriminant analysis with random matrix multiplication. *Numerical Linear Algebra with Applications*, 22(6):1180–1188, 2015.

[32] H. Ye, Y. Li, C. Chen, and Z. Zhang. Fast Fisher discriminant analysis with randomized algorithms. *Pattern Recognition*, 72:82–92, 2017.

[33] J. Ye, R. Janardan, and Q. Li. Two-dimensional linear discriminant analysis. In *Advances in Neural Information Processing Systems 17*, pages 1569–1576. MIT Press, 2005.

[34] J. Ye and Q. Li. A two-stage linear discriminant analysis via qr-decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):929–941, 2005.

[35] L. Zhang, M. Mahdavi, R. Jin, T. Yang, and S. Zhu. Random projections for classification: A recovery approach. *IEEE Transactions on Information Theory*, 60(11):7300–7316, 2014.

[36] Z. Zhang, G. Dai, C. Xu, and M. I. Jordan. Regularized discriminant analysis, ridge regression and beyond. *Journal of Machine Learning Research*, 11:2199–2228, 2010.

[37] H. Zhao and P. C. Yuen. Incremental linear discriminant analysis for face recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(1):210–221, 2008.