LINEAR DISCRIMINANT ANALYSIS WITH THE RANDOMIZED KACZMARZ METHOD

JOCELYN T. CHI† AND DEANNA NEEDELL‡

Abstract. We present a randomized Kaczmarz method for linear discriminant analysis (rkLDA), an iterative randomized approach to binary-class Gaussian model linear discriminant analysis (LDA) for very large data. We harness a least squares formulation and mobilize the stochastic gradient descent framework to obtain a randomized classifier with performance that can achieve comparable accuracy to that of full data LDA. We present analysis for the expected change in the LDA discriminant function if one employs the randomized Kaczmarz solution in lieu of the full-data least squares solution that accounts for both the Gaussian modeling assumptions on the data and algorithmic randomness. Our analysis shows how the expected change depends on quantities inherent in the data such as the scaled condition number and Frobenius norm of the input data, how well the linear model fits the data, and choices from the randomized algorithm. Our experiments demonstrate that rkLDA can offer a viable alternative to full data LDA on a range of step-sizes and numbers of iterations.

Key words. classification, stochastic optimization, supervised learning

MSC codes. 65F10, 68T10, 68W20, 68W40

1. Introduction. Linear discriminant analysis is a classic method of supervised learning, or classification. Its many applications include classification in genetics [96, 94], facial recognition [53], pattern recognition [17], hyperspectral imaging [26], and chemometrics [35]. Consider a matrix $X \in \mathbb{R}^{n \times p}$, where each row contains an observation with measurements on $p$ features. Each observation $x \in \mathbb{R}^p$ belongs to one of two classes with class label $y$ so that $y = 1$ for Group 1 and $y = 0$ for Group 2. The Gaussian model of linear discriminant analysis (LDA) assumes class-conditional distributions so that $p(x \mid y)$ is multivariate Gaussian with class means $\mu_1 \in \mathbb{R}^p$ and $\mu_2 \in \mathbb{R}^p$ and common covariance $\Sigma \in \mathbb{R}^{p \times p}$. Bayes rule then identifies a linear decision boundary for classifying observations to the most likely class.

For moderately sized data in the $n > p$ case, LDA is a classic statistical method of supervised learning. However, it encounters limitations when either $n$, or both $n$ and $p$, are very large. If $n$ is large, it may be infeasible to store the entire dataset in working memory on a local machine. If $p$ is very large, it may be very costly to perform an eigenvalue decomposition on the covariance matrix $\Sigma$, which is the computational bottleneck in performing LDA.

To address these computational limitations, we harness the least squares formulation of binary-class LDA and propose LDA with the randomized Kaczmarz method (rkLDA). The randomized Kaczmarz method (RK) [1, 13, 18, 52, 54, 55, 62, 63, 69, 70, 83, 84, 103] is an instance of stochastic gradient descent (SGD) [80] applied to a least squares objective function with a specific step-size that depends on the row chosen in a given iteration [63]. By employing a least squares formulation, we can mobilize the well-known SGD [6, 7, 66] framework to obtain a randomized LDA classifier with comparable performance to that of full data LDA. Here, “full
data” refers to the full design matrix $X$ and response $y$ without randomization or dimension reduction.

Binary classification has numerous important applications in a wide range of domains. For example, it is widely used to predict outcomes in medicine [87]. While there are numerous examples, we list two to demonstrate its wide applicability. As an example, it is important to predict whether a patient treated with a stent for coronary artery disease will suffer from in-stent restenosis, or eventual failure of the intervention to reopen an occluded artery [31]. It is also important to predict whether a treated heart failure patient will spend more than a prespecified fraction of follow-up time hospitalized versus at home; this metric has been shown to correlate well with a patient’s severity of illness [22]. Binary classification problems are also abundant in finance and banking, where it is important to accurately predict loan defaults [33, 102] and bankruptcy [39, 47, 49] for example. It also has numerous public policy applications such as in predicting nursing home admissions among the elderly [10], foster care placement [60], and food assistance program participation [44].

1.1. Contributions. While applying the RK to a least squares formulation of LDA is natural, careful investigation involves several important details. These include analysis of the expected change in the LDA discriminant function when employing the randomized Kaczmarz solution in lieu of the full data least squares solution that accounts for both algorithmic and model-induced randomness, and investigating the impact of the rkLDA solution on the optimal intercept defined in Section 2.1. Our theoretical results in Sections 4 and 5 present analysis for the expected deviation of the rkLDA discriminant function on new data from the full data one after a finite number of steps. Our numerical experiments in Section 6 demonstrate that rkLDA can offer a viable alternative to binary-class LDA.

1.2. Related works. The least squares formulation of binary-class LDA is well-known ([38, Section 4.3, Exercise 4.2], [79, Section 3.2]) and has been employed for binary sparse high dimensional discriminant analysis [58]. For more than two classes, a least squares formulation is also available for Fisher’s linear discriminant analysis (FDA) [32, 99], which omits distributional assumptions on the data and is not the same as LDA in general. Other versions of discriminant analysis employing a least squares formulation include optimal scaling [9] and flexible [38, Section 12.5] discriminant analysis, which enable non-linear transformations of the class labels.

In recent years, sketching – or dimension reduction by random sampling, random projections, or both – has become popular for large data problems. Sketching appears to have first appeared as a subspace embedding in [82]. Common categorizations of these approaches depend on whether the sketch achieves row compression [8, 14, 15, 24, 25, 41, 56, 76, 81], column compression [4, 42, 59, 86, 100], or both [61, 91].

Some works on randomized LDA and its variations exist. These focus on sketching via subspace embeddings and include feature vector compression [27], compression of the observations [50], fast Johnson-Lindenstrauss transform (FJLT) [2] projection for Fisher’s LDA [98], and random projections in LDA ensembles [28]. These utilize the well-known two-step procedure of first performing efficient dimension reduction with a suitable sketching matrix, and then solving the problem in the reduced dimension.

However, this two-step random projections approach has limited applicability in analyzing massive data. Although applying a random projection requires less computation time than dense matrix-matrix multiplication when utilizing FJLTs, it still requires $O(np \log p) + \text{poly}(\epsilon)$ time [95, page 7] when $n > p$ for a desired error $\epsilon$. Computational time can be drastically reduced if the data are sparse but we consider
the general case where the data lack known structure. Thus, the computation time required can be extraordinarily large for massive data. Perhaps one version of matrix sketching that could be combined with RK, is randomized block Kaczmarz [64, 69, 77], where not one row but a block of rows is sampled in each iteration. We discuss this as potential future work for speeding up convergence in Section 7.

Randomized versions of FDA and its regularized variants also exist [16, 88, 98]. In contrast with LDA, FDA does not involve Gaussian assumptions on the data and consequently involves a different objective function. While the FDA problem is typically solved as a generalized eigenvalue problem, it admits a least squares formulation [99] that can be employed for multiclass linear discriminant analysis. Therefore, the RK can also be applied to FDA to obtain a multiclass linear classifier.

Finally, we highlight some other state-of-the-art least squares solvers. If the design matrix is sparse or has clustered singular values, for example, the conjugate gradient method (CG) is a renowned iterative method that can converge very rapidly [34, 36, 89]. We make no such structural assumptions on the data, however, and our numerical experiments show how rkLDA can achieve a sufficiently accurate solution more rapidly than CG on some well-known classification datasets. Additionally, the RK often outperforms CG [84, Section 4.2] and requires less working memory, making it preferable when memory constraints are an issue. Another celebrated approach to solving large linear systems is to employ randomized matrix sketching as a preconditioner followed by an iterative least squares solver such as LSQR, as in [4]. While this approach yields fast and highly accurate least squares solutions, one benefit of the RK approach is that the algorithm does not need to observe the entire data. Therefore, an RK approach opens the doors to future work on a distributed rkLDA to handle data too large to fit in local memory. We discuss potential adaptations for distributed environments in Section 7.

1.3. Notation. We assume that vectors are column vectors. We denote the $j^{th}$ row of a matrix $X$ by $x_j^T$, its Frobenius norm by $\|X\|_F$, and its operator norm by $\|X\|_2$. We denote the scaled condition number of $X$ by $\kappa(X) = \|X\|_F^2\|(X^T X)^{-1}\|_2$ or $\kappa(X) = \|X\|_F^2\|(X^T X)^{\dagger}\|_2$ if $X$ lacks full column rank. This is related to the square of the scaled condition number for square matrices in [20]. Given an integer $n$, denote the set $\{1, 2, \ldots, n\}$ with $[n]$ and the all ones vector of length $n$ with $1_n$.

2. Gaussian Model Linear Discriminant Analysis (LDA). We review LDA to underscore the computational difficulties in computing an LDA classifier for very large data with $n > p$. Let $y \in \mathbb{R}^n$ contain class labels for $n$ observations, each belonging to one of $g$ classes. Let $x_i$ and $y_i$ denote the feature vector and class label for the $i^{th}$ observation. LDA assumes that observations are class-conditionally Gaussian with class means $\mu_k \in \mathbb{R}^p$ for $1 \leq k \leq g$, common covariance $\Sigma \in \mathbb{R}^{p \times p}$, and prior probability $\pi_k$ of belonging to the $k^{th}$ class with $\sum_{k=1}^{g} \pi_k = 1$. Assume that $\Sigma$ is nonsingular. Given a new observation $\tilde{x} \in \mathbb{R}^p$, Bayes rule classifies according to

\begin{equation}
\hat{y} = \arg \max_k \delta_k(\tilde{x})
\end{equation}

with the linear discriminant functions ([38, Section 4.3], [79, Section 3.1])

\begin{equation}
\delta_k(\tilde{x}) = \log \pi_k + \tilde{x}^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k \quad \text{for} \quad 1 \leq k \leq g.
\end{equation}

Since $\pi_k, \mu_k,$ and $\Sigma$ are unknown, we employ their data-dependent estimates. Let
$n_k$ denote the number of observations in the $k^{th}$ class. Define

$$\hat{\pi}_k = \frac{n_k}{n}, \quad \hat{\mu}_k = \frac{1}{n_k} \sum_{y_i=k} x_i, \text{ and } \hat{\Sigma} = \frac{1}{n-g} \sum_{k=1}^{g} \sum_{y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T.$$ 

In practice, we do not compute $\Sigma^{-1}$ explicitly. Rather than maximizing (2.2), we can equivalently minimize

$$\delta_k^*(\tilde{x}) = -\delta_k(\tilde{x}) + \frac{1}{2} \tilde{x}^T \Sigma^{-1} \tilde{x} = \frac{1}{2} (\tilde{x} - \mu_k)^T \Sigma^{-1} (\tilde{x} - \mu_k) - \log \pi_k$$

since the additive term does not depend on $k$. Let $\Sigma = VDVT$ be an eigenvalue decomposition of $\Sigma$. Since $\Sigma^{-1} = VD^{-1}VT$ exists, we write the first term in (2.3) as

$$\frac{1}{2} (\tilde{x} - \mu_k)^T \Sigma^{-1} (\tilde{x} - \mu_k) = \frac{1}{2} \|D^{-\frac{1}{2}}V(\tilde{x} - \mu_k)\|_2^2 = \frac{1}{2} \|x^* - \mu^*_k\|_2^2,$$

where $x^* = D^{-\frac{1}{2}}V^T \tilde{x}$ and $\mu^*_k = D^{-\frac{1}{2}}V^T \mu_k$. This enables rewriting (2.3) as

$$\delta_k^*(\tilde{x}) = \frac{1}{2} \|\tilde{x}^* - \mu^*_k\|_2^2 - \log \pi_k.$$ 

Applying $D^{-\frac{1}{2}}V^T$ on the left in (2.4) has a sphering effect since $D^{-\frac{1}{2}}V^T X$ has identity covariance. Thus, classifying $\tilde{x}^*$ to the group with least $\delta_k^*$ is equivalent to classifying to the nearest centroid after sphering the data and adjusting for class proportions.

Therefore, the computational burden in solving (2.1) is in forming $\Sigma$ and computing an eigenvalue decomposition for it (or a singular value decomposition for $X$). This is straightforward for moderately sized data but computationally burdensome if $n$, or both $n$ and $p$, are very large since forming $\Sigma$ requires $O(np^2)$ computations and computing its eigenvalue decomposition requires $O(p^3)$ computations [72].

### 2.1. Least Squares Formulation of LDA.

For binary classification problems, we can employ least squares regression to obtain an LDA classifier in lieu of (2.4). Therefore, we focus on this fundamental two-class problem in the $n > p$ case, where (2.1) reduces to the following: we classify $\tilde{x}$ to the second class if and only if

$$\delta_{LDA} = \left( \tilde{x} - \frac{1}{2} (\mu_1 + \mu_2) \right)^T \Sigma^{-1} (\mu_2 - \mu_1) + \log \left( \frac{\pi_2}{\pi_1} \right) > 0.$$ 

Let the $\beta^{LDA} = \Sigma^{-1} (\mu_2 - \mu_1)$ term in (2.5) denote the LDA classifier. Then the least squares solution vector is proportional to $\beta^{LS}$ ([38, Section 4.3, Exercise 4.2], [79, Section 3.2]). Following [32, Section 4], [38, Exercise 4.2], and [58, Section 3.1], we recode the class labels as $y_i = - \frac{n}{n_2}$ for observations belonging to Class 1 or $y_i = \frac{n}{n_2}$ for Class 2, $1 \leq i \leq n$. We obtain the Bayes LDA classifier by first solving

$$\left( \hat{\beta}_L, \hat{\beta}_S \right) = \arg \min_{\beta_0, \beta} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_0 - \tilde{x}_i^T \beta)^2 \right\},$$

where $\beta_0$ is called the intercept and $\hat{\beta}_o$ is the least squares estimate for $\beta_0$. Let $\hat{\beta}^{\text{opt}}_o$ denote an optimal choice for $\beta_0$. We classify $\tilde{x}$ to the second class if and only if

$$\delta_{LSLDA} = \tilde{x}^T \hat{\beta}_S + \hat{\beta}^{\text{opt}}_o > 0.$$
2.1.1. Computing the Intercept. The LDA classifier in (2.5) does not produce an intercept. Obtaining a vector $\hat{\beta}^{\text{LS}}$ that is a scalar multiple of $\hat{\beta}^{\text{LDA}}$ so that the LDA classification boundary is unchanged, however, requires fitting an intercept in (2.6). The choice of the intercept affects classification, however, and there are multiple approaches to selecting it. For example, [38, Section 4.3] recommends employing a method, such as cross-validation, that minimizes the training error for a given dataset. Such methods are sometimes referred to as data-driven approaches. Meanwhile, [58, Proposition 2] shows that the following closed-form expression for the intercept is optimal in the sense that it minimizes the expected change in the LDA discriminant function for new data

\begin{equation}
\hat{\beta}_o^{\text{opt}} = -\frac{1}{2}(\hat{\mu}_1 + \hat{\mu}_2)^{T}\hat{\beta}^{\text{LS}} + (\hat{\beta}^{\text{LS}})^{T}\Sigma\hat{\beta}^{\text{LS}}\left((\mu_2 - \mu_1)^{T}\hat{\beta}^{\text{LS}}\right)^{-1}\log\left(\frac{n_2}{n_1}\right).
\end{equation}

We employ the optimal $\hat{\beta}_o^{\text{opt}}$ from (2.8) in our experiments in Section 6. Algorithm 2.1 summarizes these procedures. While the least squares approach to LDA omits the intercept, the RK proceeds as follows. In the $k$th step, we sample the $i$th observation according to some sampling distribution $D$ and compute the $i$th update as

\begin{equation}
\beta_{k+1} = \beta_k + \frac{y_i - (x_i, \beta_k)}{\|x_i\|^2}x_i,
\end{equation}

where $c > 0$ denotes the step-size. Therefore, the RK is an instance of weighted SGD where the objective function is the least squares objective [63]. If $c = 1$ and the rows are sampled with probabilities proportional to their row norms, the RK converges exponentially to a solution that is within a radius of $\hat{\beta}^{\text{LS}}$ [84]. Other works have shown that appropriate choices of $c < 1$ can enable convergence within this radius [12, 37, 65, 85, 93]. Finite-iteration guarantees for $n > p$ show a trade-off between the size of the radius and the convergence rate for any choice of $c < 1$ in (3.1) [63].
3.0.1 Importance Sampling. Selecting rows in (3.1) with probabilities proportional to their row norms as in [84] is an example of importance sampling, where greater sampling weights may be placed on some observations. Importance sampling [63, 67, 78, 84] in (3.1) with any normalized sampling weight \(w_i \geq 0\) such that \(\mathbb{E}_{i \sim D}[w_i] = 1\) can be cast in the weighted SGD framework. If the Lipschitz constants of the SGD gradient estimates have substantially different values, this is preferable to uniform sampling, where the rows enjoy equal weights [63, Section 2].

A common variation of importance sampling is leverage score sampling, which appears in sketched least squares works [14, 15, 23, 41, 56, 74]. Given a matrix \(\mathbf{X} \in \mathbb{R}^{n \times p}\) with thin singular value decomposition \(\mathbf{X} = UDV^T\), its row leverage scores are given by the squared Euclidean norms of the rows of \(U\): \(\ell_j = \|u_j^T\|^2_2\) for \(1 \leq j \leq n\). Leverage scores satisfy \(0 \leq \ell_j \leq 1\) and \(\sum_{j=1}^n \ell_j = p\) [23].

4. Randomized Kaczmarz LDA. We harness the least squares formulation of binary-class LDA and propose randomized Kaczmarz LDA (rkLDA). Algorithm 4.1 presents rkLDA with importance sampling. Assume i.i.d. weighted random sampling for selecting row \(i = i_k\) at the \(k\)th iteration of (3.1) with probability \(p_i\) for \(1 \leq i \leq n\) such that \(\sum_{i=1}^n p_i = 1\). Convergence guarantees for the RK iterates [63, Corollary 5.1, Corollary 5.2, Corollary 5.3] hold for the rkLDA iterates \(\beta_k\) conditioned on \(\mathbf{X}\).

Algorithm 4.1 Randomized Kaczmarz LDA

**Input:** Labeled data \((\mathbf{y} \in \mathbb{R}^n, \mathbf{X} \in \mathbb{R}^{n \times p})\) where the observations belong to one of two classes, new unlabeled observation \(\tilde{x} \in \mathbb{R}^p\), initial \(\beta_0 \in \mathbb{R}^{p+1}\), maximum iterations \(K \in \mathbb{N}\), sampling probabilities \(p_i\) for \(1 \leq i \leq n\) with \(\sum_{i} p_i = 1\), and step-size \(c < 1\)

**Output:** Predicted class label \(\hat{y}\) for \(\tilde{x}\)

1: Recode class labels as \(y_i = \frac{\mathbf{y}_i}{n_1}\) for Class 1 and \(y_i = \frac{\mathbf{y}_i}{n_2}\) for Class 2 for \(1 \leq i \leq n\)
2: Compute centroid estimates \(\mu_1\) and \(\mu_2\):
   \[\mu_k = \frac{1}{n_k} \sum_{y_i=k} \mathbf{x}_i\] for \(k = 1, 2\)
3: \(\mathbf{X_c} = (\mathbf{1}_n \mathbf{X})\) // Append \(\mathbf{1}_n\) to estimate \(\beta_0\)
4: For \(k = 0, 1, 2, \ldots, K-1\):
5: Randomly sample \(i\) from \([n]\) with probability \(p_i\)
6: \(\beta_{k+1} = \beta_k + c \cdot \frac{y_i - \langle \mathbf{x}_i, \beta_k \rangle}{\|\mathbf{x}_i\|^2} \mathbf{x}_i\)
7: Set \(\beta_{k+1}^{\text{opt}}\) to be the vector obtained from the last \(p\) entries of \(\beta_K\)
8: Select \(\beta_{k+1}^{\text{opt}}\) via (2.8)
9: Classify \(\tilde{x}\) to Class 2 if and only if \(\tilde{x}^T \hat{\beta}_{\text{RK}} + \tilde{x}^T \hat{\beta}_{\text{opt}} > 0\)
10: return Predicted class label \(\hat{y}\) for \(\tilde{x}\)

Each iteration in Algorithm 4.1 requires \(O(p)\) flops. Runtime experiments in Section 6.3 give empirical evidence of the number of iterations required in the context of classification.

4.1. Quantifying the expected difference between the rkLDA and full data LDA discriminant functions on new data. Consider new unlabeled data \(\mathbf{X} \in \mathbb{R}^{N \times p}\). A primary question is: How does the LDA discriminant function on new data \(\mathbf{X}\) change if we employ the RK solution \(\hat{\beta}_{\text{RK}}\) from Algorithm 4.1 in place of the full data solution \(\beta_{\text{LS}}\) from Algorithm 2.1? To answer this question, we consider three sources of randomness: 1) algorithmic randomness from the RK iterates \(\{i\}\), which are sampled according to some sampling distribution \(D\); 2) model-induced randomness of the training data – the class-conditional distribution of the training data \(\mathbf{X}\); and
3) model-induced randomness of the new data – the class-conditional distribution of the new unlabeled data $\tilde{X}$. Therefore, we focus on the expected change in the LDA discriminant function conditioned on the training data at the $k^{th}$ step of Algorithm 4.1: $E \left[ \| \tilde{X} \beta_k - \tilde{X} \beta^{LS} \|^2 \mid X \right]$. The conditional expectation is with respect to both algorithmic randomness from RK and model-induced randomness of the new data.

Theorem 4.1 presents the expected change from the last $p$ entries in $\tilde{X} \beta_k$ and $\tilde{X} \beta^{LS}$ for any sampling probability $p_i$ such that $\sum_{i=1}^n p_i = 1$. These are estimated with the intercept term included in the regression model but do not include the optimal intercept since it is typically computed via a separate post-processing as described in Section 2.1.1. Section 4.2 discusses the expected change in $\hat{\beta}^{opt}$ from (2.8) if estimated with $\hat{\beta}^{RK}$ rather than $\hat{\beta}^{LS}$. For notational simplicity, let the first column of the training data $X \in \mathbb{R}^{n \times (p+1)}$ be $1_n$.

Theorem 4.1. Given training data $X \in \mathbb{R}^{n \times (p+1)}$, where the first column of $X$ is $1_n$, and recorded class labels $y \in \mathbb{R}^n$ as in Algorithm 4.1, let $\beta$ be the minimizer of $\hat{\beta} = \arg \min_{\beta} \frac{1}{2} \| X \beta - y \|^2$, and let $\beta_k$ and $\beta^{LS}$ be the vectors formed from the last $p$ entries in $\beta_k$ from Algorithm 4.1 and from $\hat{\beta}$, respectively. Let $p_i$ for $1 \leq i \leq n$ with $\sum_{i=1}^n p_i = 1$ denote sampling probabilities in Algorithm 4.1. For notational simplicity, let $\alpha_i = \frac{\| x_i \|^2}{p_i}$ for $1 \leq i \leq n$ and let $\hat{\alpha}$ be a lower bound for $\alpha_i$ so that $\hat{\alpha} \leq \alpha_i \leq \sup_i \alpha_i = \| X \|^2_F$ almost surely. For any new unlabeled data $\tilde{X} \in \mathbb{R}^{N \times p}$, $\tilde{X} \beta_k$ from Algorithm 4.1, where the $i^{th}$ observation is selected at the $k^{th}$ step with probability $p_i$, satisfies

$$E \left[ \| \tilde{X} \beta_k - \tilde{X} \beta^{LS} \|^2 \mid X \right] \leq \left( 1 - 2 \frac{c}{n(X)} \left( 1 - \frac{c}{\hat{\alpha}} \| X \|^2_F \right) \right)^k E \| \tilde{X} \|^2 \| \hat{\beta} - \hat{\beta}^{LS} \|^2 \| X \|^2_F \| X \|^2 r^*,$$

$$+ \frac{c}{\hat{\alpha}} \left( 1 - \frac{c}{\hat{\alpha}} \| X \|^2_F \right) E \| \tilde{X} \|^2 r^*,$$

(4.1)

where $c < \frac{\alpha_i}{\sup p_i \alpha_i} < 1$ is a fixed step-size, and $r^* = \| X \beta - y \|^2$. The conditional expectation is with respect to the sampling distribution of the iterates $\{ i \}$ and the class-conditional distribution of $\tilde{X}$.

Theorem 4.1 shows the expected change in the LDA discriminant function if we employ the RK solution in lieu of the full-data least squares solution. In particular, Theorem 4.1 shows how this expected difference depends on the following: 1) quantities that depend on the data such as the conditioning of $X$, 2) how well the linear model fits the data (as quantified by $r^*$), and 3) choices in the RK algorithm such as the step-size $c$, the sampling probabilities $p_i$, and the number of iterates $K$.

The last term in (4.1) contains the radius, or convergence horizon, described in Section 1. This horizon is positive given the definitions of $\hat{\alpha}$ and $c$. Additionally, the dependence on $c$ enables a trade-off between the size of the horizon and convergence speed: a smaller $c$ enables a smaller horizon but also results in slower convergence. This radius depends on elements that are independent of the algorithm for any given dataset with labels $y$, training data $X$, and new unlabeled data $\tilde{X}$. For example, the size of the radius depends on how close $y$ is to the column space of $X$ as indicated by the residual $r^*$ and the conditioning of $X$. In particular, we observe how Theorem 4.1 depends on the conditioning of $X$. Notice that from [84, (3)] we obtain $1 \leq \frac{\alpha_i}{p_i} \leq k^2(X)$, where $k(X)$ is the traditional condition number defined by the ratio of the largest singular value of $X$ to its smallest singular value. If $X$ is
perfectly conditioned so that $k(X) = 1$, then $\kappa(X) = p + 1$. Moreover, $\kappa(X)$ increases as conditioning worsens. Therefore, the bound in Theorem 4.1 is larger when $X$ is poorly conditioned and is smaller when $X$ is well-conditioned.

We next observe the number of iterations $k$ required to achieve an expected discriminant function error tolerance $\epsilon$. This follows directly from Theorem 4.1 and holds for any desired tolerance $\epsilon > 0$ and suitably chosen sampling probabilities $p_i$.

**Corollary 4.2.** Given a tolerance $0 < \epsilon < \frac{1}{2}$, a step-size of

$$c = \frac{\epsilon \hat{\alpha}}{2 \kappa(X) E \|X\|_2^2 + 2 \epsilon \|X\|_F^2}$$

is sufficient to ensure that after

$$k \geq \log \left( \frac{2 \epsilon_0 E \|\bar{X}\|_2^2}{\epsilon} \left( \frac{\|X\|_2^2}{4 E \|X\|_2^2} + \frac{\kappa(X)}{2\epsilon} \right) \right)$$

iterations in Algorithm 4.1, we have $E \|\bar{X}\beta_k - \bar{X}\beta^L_S\|_2 \leq \epsilon (1 + r^*)$, where $\epsilon_0 = \|\beta_0 - \beta^L_S\|_2^2$, $r^* = \|\bar{X}\beta - y\|_2^2$, and the expectation is over the random sampling in Algorithm 4.1 as well as the randomness in $\bar{X}$.

Compared with [63, Theorem 2.1], which holds generally for any $\mu$-strongly convex function, Theorem 4.1 focuses on the least squares formulation for binary-class LDA in (2.6). Since LDA is a classification problem, we focus on $\bar{X}\beta_k$ on new unlabeled data at the $k$th iterate rather than the $k$th iterate $\hat{\beta}_k$. Additionally, Theorem 4.1 accounts for the Gaussian modeling assumptions on the training data $X$ and unlabeled data $\bar{X}$. Therefore, the conditional expectation in Theorem 4.1 is with respect to the model-induced randomness of $\bar{X}$ after accounting for the algorithm-induced randomness from the iterates in (3.1). Since we account for randomness in $\bar{X}$, the first term in Theorem 4.1 depends additionally on $E \|\bar{X}\|_2^2$.

Since many results in this setting are “worst-case” results and often overly pessimistic, in theory, a large $k$ might be required to guarantee a very small expected discriminant function error. In practice, however, our numerical experiments demonstrate that very good classification results can be obtained even when $k$ is quite small. In our real data experiments with larger $p$ in Section 6.3, for example, fewer than 2,500 iterations were required to obtain performance that was very comparable to LDA.

We make some observations about $E \|\bar{X}\|_2$, the expected largest singular value of new data $\bar{X}$. For large $N$ and $p$ and standardized data such that the entries of $\bar{X}$ are i.i.d. $\mathcal{N}(0, 1)$, one can employ asymptotic results on the largest singular value of $\bar{X}$ from [29, Proposition 6.1]. Namely, if the entries of $\bar{X}$ are i.i.d. $\mathcal{N}(0, 1)$, and $N$ and $p$ tend to infinity such that $\frac{N}{p}$ tends to a limit $d \in [0, 1]$, then $\frac{1}{\sqrt{N}} \|\bar{X}\|_2$ converges to $1 + \sqrt{d}$ almost surely. It follows that if we additionally have that $N = p$ so that $d = 1$, then $\|\bar{X}\|_2$ converges to $2\sqrt{N}$ almost surely. Additionally, if $N = p$ and the entries of $\bar{X}$ are i.i.d. $\mathcal{N}(0, \frac{1}{p})$, then [19, Theorem II.4] offers the following bounds on $\|\bar{X}\|_2$

$$\mathcal{P} \left( \|\bar{X}\|_2 \geq s_u \right) \leq \exp(-\gamma s_u^2 N^2) \quad \text{and} \quad \mathcal{P} \left( \|\bar{X}\|_2 \leq s_t \right) \leq (Cs_t)^{N^2},$$

where $s_u \leq s_0, s_t \geq 0$, and $\gamma, C$, and $s_0$ are universal positive constants. These bounds provide a streamlined view of how the largest singular value of the new data $\bar{X}$ depends on $N$ under some simplified conditions.
The following corollaries illustrate Theorem 4.1 for some common choices of sampling probabilities $p_i$. Table 1 at the end of this section summarizes their differences. Corollary 4.3 holds for row weight sampling, where $p_i = \frac{\|x_i\|^2}{\|X\|^2}$ ([63, Corollary 5.1] and [84, Section 2]).

**Corollary 4.3.** Given training data $X \in \mathbb{R}^{n \times (p+1)}$, where the first column of $X$ is $1_n$, and recoded class labels $y \in \mathbb{R}^n$ as in Algorithm 4.1, let $\hat{\beta}$ be the minimizer of $\beta = \arg\min_{\beta} \frac{1}{2} \|X\beta - y\|^2_2$, and let $\hat{\beta}_k$ and $\hat{\beta}^{LS}_k$ be the vectors formed from the last $p$ entries in $\hat{\beta}_k$ from Algorithm 4.1 and from $\hat{\beta}$, respectively. Let $c < 1$. For any new unlabeled data $\tilde{X} \in \mathbb{R}^{N \times p}$, $\tilde{X}\hat{\beta}_k$ from Algorithm 4.1, where the $i^{th}$ observation is selected at the $k^{th}$ step with probability $p_i = \frac{\|x_i\|^2}{\|X\|^2}$ satisfies

\[
\mathbb{E}\left[\left\|\tilde{X}\hat{\beta}_k - \tilde{X}\hat{\beta}^{LS}_k\right\|^2_2 \mid X\right] \leq \left(1 - 2c\left(1 - c\right)\kappa(X)\right)^k \mathbb{E}\left[\left\|\tilde{X}\right\|^2_2 \left\|\hat{\beta}_0 - \hat{\beta}^{LS}\right\|^2_2\right] + \frac{c}{1 - c}\kappa(X) \mathbb{E}\left[\left\|\tilde{X}\right\|^2_2 r\right],
\]

where $r = \frac{\|X\hat{\beta} - Xy\|^2_2}{\|X\|^2}$. The conditional expectation is with respect to the sampling distribution of the iterates $\{i\}$ and the class-conditional distribution of $\tilde{X}$.

Similar to [63, Corollary 5.1] and [84, Theorem 2], Corollary 4.3 does not depend on the number of observations. Rather, convergence with row weight sampling depends on the conditioning of $X$, the distance between the initial estimate $\hat{\beta}_0$ and the minimum $\hat{\beta}^{LS}$, the least squares residuals at the minimum in $r$, and $\mathbb{E}\left[\left\|\tilde{X}\right\|^2_2\right]$. Notice that if $y$ is close to the column space of $X$ so that the least squares residuals in $r$ are small, then the convergence horizon in (4.3) is likewise smaller. For very large $n$, pre-computing the row norms may require batched computations on subsets of the rows of the training data. For comparison, Corollary 4.4 illustrates Theorem 4.1 with uniform sampling, where $p_i = \frac{1}{n}$ so that all rows are selected with equal probabilities.

**Corollary 4.4.** Given training data $X \in \mathbb{R}^{n \times (p+1)}$, where the first column of $X$ is $1_n$, and recoded class labels $y \in \mathbb{R}^n$ as in Algorithm 4.1, let $\beta$ be the minimizer of $\beta = \arg\min_{\beta} \frac{1}{2} \|X\beta - y\|^2_2$, and let $\hat{\beta}_k$ and $\hat{\beta}^{LS}_k$ be the vectors formed from the last $p$ entries in $\hat{\beta}_k$ from Algorithm 4.1 and from $\beta$, respectively. Let $c < \min\left(1, \frac{n}{\kappa(X)}\right)$. For any new unlabeled data $\tilde{X} \in \mathbb{R}^{N \times p}$, $\tilde{X}\hat{\beta}_k$ from Algorithm 4.1, where the $i^{th}$ observation is selected at the $k^{th}$ step with probability $p_i = \frac{1}{n}$ satisfies

\[
\mathbb{E}\left[\left\|\tilde{X}\hat{\beta}_k - \tilde{X}\hat{\beta}^{LS}_k\right\|^2_2 \mid X\right] \leq \left(1 - 2c\left(1 - c\right)\kappa(X)\right)^k \mathbb{E}\left[\left\|\tilde{X}\right\|^2_2 \left\|\hat{\beta}_0 - \hat{\beta}^{LS}\right\|^2_2\right] + \frac{c}{1 - c}\kappa(X) \mathbb{E}\left[\left\|\tilde{X}\right\|^2_2 r\right],
\]

where $r = \frac{\|X\hat{\beta} - Xy\|^2_2}{\|X\|^2}$. The conditional expectation is with respect to the sampling distribution of the iterates $\{i\}$ and the class-conditional distribution of $\tilde{X}$.

Compared with [63, Corollary 5.2], which employs a diagonal scaling matrix $D$ whose $i^{th}$ diagonal entry is $\|x_i\|^2_2$, Corollary 4.4 utilizes uniform sampling probabilities $p_i = \frac{1}{n}$. This highlights the increased dependency on the conditioning of $X$: compared with Corollary 4.3, Corollary 4.4 depends on $\kappa^2(X)$ rather than $\kappa(X)$. If we employ
a diagonal scaling matrix $D$ as in [63, Corollary 5.2], we likewise observe exponential convergence to a radius of the weighted least squares solution scaled by $E \| \hat{X} \|_2^2$.

Finally, since leverage score sampling is a common form of importance sampling in many sketching problems, Corollary 4.5 illustrates Theorem 4.1 for leverage score sampling with $p_i = \frac{\ell_i}{p}$. While leverage score sampling is not recommended for massive unstructured matrices, our results hold for all data – not only massive structured data. Therefore, since leverage score sampling appears frequently in the matrix sketching literature, we include the following result to illustrate the flexibility of Theorem 4.1.

**Corollary 4.5.** Given training data $X \in \mathbb{R}^{n \times (p+1)}$, where the first column of $X$ is $1_n$, and recorded class labels $y \in \mathbb{R}^n$ as in Algorithm 4.1, let $\beta$ be the minimizer of $\beta = \arg \min_\beta \frac{1}{2} \| X \beta - y \|_2^2$ and let $\hat{\beta}_k$ and $\hat{\beta}_{LS}$ be the vectors formed from the last $p$ entries in $\beta_k$ from Algorithm 4.1 and from $\hat{\beta}$, respectively. Let $c < \frac{p}{\kappa(X)}$ and let $\ell_i$ denote the $i$th row leverage score of the training data computed from the original $p$ variables only. Then for any new unlabeled data $X \in \mathbb{R}^{N \times p}$, $\hat{X}\hat{\beta}_k$ from Algorithm 4.1, where the $i$th observation is selected at the $k$th step with probability $p_i = \frac{\ell_i}{p}$, satisfies

$$
E \left[ \| \hat{X}\hat{\beta}_k - \hat{X}\hat{\beta}_{LS} \|_2^2 \mid X \right] \leq \left( 1 - 2c \frac{1}{\kappa(X)} - \frac{c}{p} \right) \| \hat{X} \|_2^2 \| \hat{\beta}_0 - \hat{\beta}_{LS} \|_2^2 + \frac{p}{\kappa(X)} \kappa(X) \| \hat{X} \|_2^2 r,
$$

where $r = \| X \hat{\beta} - y \|_2^2 / \| X \|_F^2$. The conditional expectation is with respect to the sampling distribution of the iterates $\{\ell_i\}$ and the class-conditional distribution of $X$.

Here, we employ the fact that if $p_i = \frac{\ell_i}{p}$, then $\hat{\alpha} = \frac{1}{\| X \|_F^2}$ is a lower bound for the $\alpha_i$ to obtain the result. Similar to Corollaries 4.3 and 4.4, Corollary 4.5 does not depend on $n$. However, the leverage score probabilities introduce a dependency on the number of variables $p$. Finally, similar to Corollary 4.4, Corollary 4.5 also depends on $\kappa^2(X)$ rather than $\kappa(X)$.

For notational simplicity, we do not distinguish whether the $p_i$ in Corollaries 4.3 and 4.5 are computed with $1_n$ included in the training data $X$. However, it appears reasonable to compute the $p_i$ with the original $p$ variables only. Therefore, we adopt this approach in our numerical experiments in Section 6. Let $A = 2 \frac{\kappa(X)}{\alpha(1-\frac{c}{p})} \left( 1 - \frac{c}{p} \| X \|_F^2 \right)$ and $B = \frac{c}{\alpha(1-\frac{c}{p})}$ from the convergence rate and horizon in Theorem 4.1, respectively.

Table 1 summarizes differences in Corollaries 4.3 - 4.5.

| Sampling approach       | $p_i$ | A                     | B                     |
|-------------------------|-------|-----------------------|-----------------------|
| Row-weight sampling     | $\| X \|_2^2 / \| X \|_F^2$ | $2c \left( \frac{1}{\alpha(X)} - \frac{c}{\kappa(X)} \right)$ | $\frac{c}{1-c}$ |
| Uniform sampling        | $1/n$ | $2c \left( \frac{1}{\kappa(X)} - \frac{c}{n} \right)$ | $\kappa(X)$ |
| Leverage score sampling | $\ell_i/p$ | $2c \left( \frac{1}{\kappa(X)} - \frac{c}{p} \right)$ | $\frac{2}{p} \kappa(X)$ |

Table 1: Table summarizing differences in Corollaries 4.3 - 4.5.

**4.2. Change in optimal intercept due to randomized algorithm.** We consider the change in $\beta_{opt}^n$ from (2.8) if one employs $\beta_{RK}$ in lieu of $\beta_{LS}$. Denote this
change by $\Delta \hat{\beta}_0^{\text{opt}}$ and observe that

$$
\Delta \hat{\beta}_0^{\text{opt}} = -\frac{1}{2}(\hat{\mu}_1 + \hat{\mu}_2)^T(\hat{\beta}_L^\text{LS} - \hat{\beta}_R^\text{RK}) + (\hat{\beta}_L^\text{LS} - \hat{\beta}_R^\text{RK})^T \Sigma(\hat{\beta}_L^\text{LS} - \hat{\beta}_R^\text{RK})\{(\hat{\mu}_2 - \hat{\mu}_1)^T(\hat{\beta}_L^\text{LS} - \hat{\beta}_R^\text{RK})\}^{-1} \log \left(\frac{n_2}{n_1}\right).
$$

Enveloping scalars inside Euclidean norms, applying the triangle and Cauchy-Schwarz inequalities, and the eigenvalue decomposition of $\Sigma$ produces

$$
\Delta \hat{\beta}_0^{\text{opt}} \leq \frac{1}{2}\|\hat{\mu}_1 + \hat{\mu}_2\|\|\hat{\beta}_L^\text{LS} - \hat{\beta}_R^\text{RK}\| + \frac{\hat{\sigma}_1^2(X)\|\hat{\beta}_L^\text{LS} - \hat{\beta}_R^\text{RK}\|^2}{\|\hat{\mu}_2 - \hat{\mu}_1\|^T(\hat{\beta}_L^\text{LS} - \hat{\beta}_R^\text{RK})\|} \log \left(\frac{n_2}{n_1}\right),
$$

where $\hat{\sigma}_1(X)$ is the largest singular value of $X$. Applying the conditional expectation with respect to $X$ on both sides produces

$$
\mathbb{E}[\Delta \hat{\beta}_0^{\text{opt}} \mid X] \leq \frac{1}{2}\|\hat{\mu}_1 + \hat{\mu}_2\|\mathbb{E}[\|\hat{\beta}_L^\text{LS} - \hat{\beta}_R^\text{RK}\| \mid X] + \hat{\sigma}_1^2(X) \mathbb{E}\left[\frac{\|\hat{\beta}_L^\text{LS} - \hat{\beta}_R^\text{RK}\|^2}{\|\hat{\mu}_2 - \hat{\mu}_1\|^T(\hat{\beta}_L^\text{LS} - \hat{\beta}_R^\text{RK})\|} \mid X\right].
$$

Theorem 2.1 from [63] shows that the first term on the right in (4.5) can be made small with sufficiently many $k$ for any given tolerance $\epsilon > 0$ and suitably chosen $p_i$. For this reason, we likewise expect the numerator in the expectation of the second term to decrease faster than the denominator. Of course, this requires that the class means are sufficiently separated so that the data are suitable for LDA and that $(\hat{\mu}_2 - \hat{\mu}_1)$ and $(\hat{\beta}_L^\text{LS} - \hat{\beta}_R^\text{RK})$ are not orthogonal. Therefore, under some mild assumptions, it appears that the expected change in the optimal intercept from (2.8) conditioned on $X$ can likewise be made suitably small.

5. Proofs of Main Results. In this section, we present proofs of the main results. We begin by restating [63, Lemma 8.1].

**Lemma 5.1** (Definition 2.1 from [101], Lemma 8.1 from [63](Co-coercivity)). For a smooth function $f$ whose gradient has Lipschitz constant $L$,

$$
\|\nabla f(x) - \nabla f(y)\|^2 \leq L(x - y, \nabla f(x) - \nabla f(y)).
$$

5.1. Proof of Theorem 4.1. We first present a general result in the SGD framework. We then tie this result to the RK by updating the SGD components via the reweighted SGD framework and the corresponding sampling weights as in [63]. Following the notation in [63, Section 2], we seek to minimize a strongly convex function $F$, where $F(\beta) = \mathbb{E}_{i \sim D} f_i(\beta)$ and each $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ is convex with Lipschitz constant $\inf L \leq L_i \leq \sup L$ almost surely. Here, the expectation is with respect to the sampling distribution of the SGD iterates $\{i\}$ according to some source distribution $D$, where the $i^{th}$ row of $X$ is selected at the $k^{th}$ iterate. Let $\bar{L}$ denote the average $L_i$. Let $\hat{\sigma}_2^2 = \mathbb{E}[\|\nabla f_i(\beta)\| X, \bar{X}]^2$ denote the expected norm of the residual gradient at the minimizer $\hat{\beta}_0 = \arg\min_\beta F(\beta)$ conditioned on the data $X$ and $\bar{X}$, and let $\gamma < \frac{1}{\sup L}$ be a fixed step-size.

For any new unlabeled data matrix $\bar{X} \in \mathbb{R}^{N \times (p+1)}$ and the $k+1^{th}$ iterate, we employ the definition of the operator norm and the SGD iterate update rule
\(\beta_{k+1} \leftarrow \beta_k - \gamma \nabla f_i(\beta_k)\) to obtain
\[
\|\tilde{X}\beta_{k+1} - \tilde{X}\beta_*\|_2^2 \leq \|\tilde{X}\|_2^2 \|\beta_k - \beta_* - \gamma \nabla f_i(\beta_k)\|_2^2
\]
(5.1)
\[
= \|\tilde{X}\|_2^2 \left(\|\beta_k - \beta_*\|_2^2 - 2\gamma \langle \beta_k - \beta_* , \nabla f_i(\beta_k) \rangle + \gamma^2 \|\nabla f_i(\beta_k)\|_2^2 \right).
\]
Focusing on the last term in (5.1), we add and subtract \(\nabla f_i(\beta_*)\) inside the norm, scale terms inside the norm by \(2 \cdot \frac{1}{2}\), and simplify to obtain
\[
\gamma^2 \|\nabla f_i(\beta_k)\|_2^2 = 4\gamma^2 \left\{ \frac{1}{2} \left(\nabla f_i(\beta_k) - \nabla f_i(\beta_*)\right) + \frac{1}{2} \nabla f_i(\beta_*) \right\}_2^2
\]
(5.2)
\[
\leq 2\gamma^2 \|\nabla f_i(\beta_k) - \nabla f_i(\beta_*)\|_2^2 + 2\gamma^2 \|\nabla f_i(\beta_*)\|_2^2,
\]
where we apply Jensen’s inequality to obtain the inequality in the second line. Employing (5.2) in (5.1) then produces
\[
\|\tilde{X}\beta_{k+1} - \tilde{X}\beta_*\|_2^2 \leq \|\tilde{X}\|_2^2 \left(\|\beta_k - \beta_*\|_2^2 - 2\gamma \langle \beta_k - \beta_* , \nabla f_i(\beta_k) \rangle + 2\gamma^2 \|\nabla f_i(\beta_* )\|_2^2 \right).
\]
Applying Lemma 5.1 to \(\|\nabla f_i(\beta_k) - \nabla f_i(\beta_*)\|_2^2\) then gives us
\[
\|\tilde{X}\beta_{k+1} - \tilde{X}\beta_*\|_2^2 \leq \|\tilde{X}\|_2^2 \left(\|\beta_k - \beta_*\|_2^2 - 2\gamma \langle \beta_k - \beta_* , \nabla f_i(\beta_k) \rangle + 2\gamma^2 \|\nabla f_i(\beta_* )\|_2^2 \right)
\]
\[
+2\gamma^2 \|\tilde{X}\|_2^2 I_i \langle \beta_k - \beta_* , \nabla f_i(\beta_k) - \nabla f_i(\beta_* ) \rangle + 2\gamma^2 \|\tilde{X}\|_2^2 \|\nabla f_i(\beta_*)\|_2^2.
\]
Employing the law of iterated expectations, we first take the expectation on both sides with respect to the sampling distribution of \(\{i\}\) while conditioning on both the training data \(X\) and the new data \(\tilde{X}\). We employ the facts that \(F(\beta) = \mathbb{E}_{i \sim D}[f_i(\beta) | X, \tilde{X}]\), \(\mathbb{E}_{i \sim D}[\nabla f_i(\beta) | X, \tilde{X}] = \nabla F(\beta), \sigma^2 = \mathbb{E}_{i \sim D}[\|\nabla f_i(\beta_*)\|^2 | X, \tilde{X}]^2\), and \(L_i \leq \sup L\) almost surely to obtain
\[
\mathbb{E}_{i \sim D} \left[\|\tilde{X}\beta_{k+1} - \tilde{X}\beta_*\|_2^2 | X, \tilde{X}\right] \leq \|\tilde{X}\|_2^2 \|\beta_k - \beta_*\|_2^2 - 2\gamma \|\tilde{X}\|_2^2 \langle \beta_k - \beta_* , \nabla F(\beta) \rangle
\]
\[
+2\gamma^2 \|\tilde{X}\|_2^2 \sup L \langle \beta_k - \beta_* , \nabla F(\beta) - \nabla F(\beta_*) \rangle + 2\gamma^2 \|\tilde{X}\|_2^2 \sigma^2.
\]
Since \(\beta_*\) minimizes \(F\), we have \(\nabla F(\beta_*) = 0\) so that
\[
\mathbb{E}_{i \sim D} \left[\|\tilde{X}\beta_{k+1} - \tilde{X}\beta_*\|_2^2 | X, \tilde{X}\right] \leq -2\gamma \|\tilde{X}\|_2^2 \|\beta_k - \beta_*\|_2^2 + 2\gamma^2 \|\tilde{X}\|_2^2 \sigma^2.
\]
(5.3)
\[
-\langle \beta_k - \beta_* , \nabla F(\beta_k) - \nabla F(\beta_*) \rangle \leq -\mu \|\beta_k - \beta_*\|_2^2.
\]
We employ the fact that \(F\) is \(\mu\)-strongly convex so that for \(\mu > 0\) we have
\[
(5.4)
-\langle \beta_k - \beta_* , \nabla F(\beta_k) - \nabla F(\beta_*) \rangle \leq -\mu \|\beta_k - \beta_*\|_2^2.
\]
Therefore, utilizing (5.4) in (5.3) gives us
\[
\mathbb{E}_{i \sim D} \left[\|\tilde{X}\beta_{k+1} - \tilde{X}\beta_*\|_2^2 | X, \tilde{X}\right] \leq \|\tilde{X}\|_2^2 \left[1 - 2\gamma \mu (1 - \gamma \sup L) \right] \|\beta_k - \beta_*\|_2^2 + 2\gamma^2 \|\tilde{X}\|_2^2 \sigma^2
\]
since \(\gamma \leq \frac{1}{\sup L}\). Now taking the expectation on both sides with respect to the class-conditional distribution of the new data \(\tilde{X}\) gives us
\[
\mathbb{E}_{\tilde{X}} \left[\|\tilde{X}\beta_{k+1} - \tilde{X}\beta_*\|_2^2 | X \right] \leq \left[1 - 2\gamma \mu (1 - \gamma \sup L) \right] \|\beta_k - \beta_*\|_2^2 \mathbb{E} \|\tilde{X}\|_2^2 + 2\gamma^2 \sigma^2 \mathbb{E} \|\tilde{X}\|_2^2.
Since we employed \( \|\tilde{X}\beta_{k+1} - \tilde{X}\beta_*\|_2^2 \leq \|\tilde{X}\|_2^2 \|\beta_{k+1} - \beta_*\|_2^2 \) in (5.1), it follows that
\[
\mathbb{E}_{\tilde{X}} \left[ \|\tilde{X}\beta_{k+1} - \tilde{X}\beta_*\|_2^2 | \tilde{X} \right] \leq \mathbb{E}_{\tilde{X}} \left[ \|\tilde{X}\|_2^2 \|\beta_{k+1} - \beta_*\|_2^2 | \tilde{X} \right] 
\leq \left[ 1 - 2\gamma \mu (1 - \gamma \text{sup} L) \right] \|\beta_{k+1} - \beta_*\|_2^2 + 2\gamma^2 \sigma^2 \mathbb{E} \|\tilde{X}\|_2^2
\]
above. After dividing both sides by \( \mathbb{E} \|\tilde{X}\|_2^2 \), we recursively apply this bound over the previous \( k \) iterations to obtain
\[
\mathbb{E} \left[ \|\tilde{X}\beta_{k+1} - \tilde{X}\beta_*\|_2^2 | \tilde{X} \right] \leq \left( 1 - 2\gamma \mu (1 - \gamma \text{sup} L) \right)^k \|\beta_0 - \beta_*\|_2^2 
+ 2 \sum_{j=0}^{k-1} \left( 1 - 2\gamma \mu (1 - \gamma \text{sup} L) \right)^j \gamma^2 \sigma^2 
\leq \left( 1 - 2\gamma \mu (1 - \gamma \text{sup} L) \right)^k \|\beta_0 - \beta_*\|_2^2 + \frac{\gamma \sigma^2}{\mu (1 - \gamma \text{sup} L)}. 
\]
(5.5)

This gives us the following general SGD result
\[
\mathbb{E} \left[ \|\tilde{X}\beta_{k} - \tilde{X}\beta_*\|_2^2 | \tilde{X} \right] \leq \left( 1 - 2\gamma \mu (1 - \gamma \text{sup} L) \right)^k \mathbb{E} \|\tilde{X}\|_2^2 \|\beta_0 - \beta_*\|_2^2 + 
\frac{\gamma \sigma^2}{\mu (1 - \gamma \text{sup} L)} \mathbb{E} \|\tilde{X}\|_2^2. 
\]
(5.6)

We update this general result for the least squares problem via randomized Kaczmarz with importance sampling [63, Section 5]. In particular, we first begin with the unweighted SGD problem arg \( \min_{\beta} F(\beta) = \arg \min_{\beta} \frac{1}{2} \|X\beta - y\|_2^2 \), where \( F \) is \( \mu \)-strongly convex with \( \mu = \frac{1}{\text{sup} \|X\|_2^2} \) and the solution is \( \beta = \arg \min_{\beta} F(\beta) \). Therefore, \( f_i = \frac{1}{2} (x_i^T \beta - y_i)^2 \) is convex with Lipschitz constant \( L_i = n \|x_i\|_2^2 \), and \( \nabla f_i = n (x_i^T \beta - y_i) x_i \). Let \( \alpha_i = \frac{\|x_i\|_2^2}{p_i} \) and let \( \tilde{\alpha} \) be a lower bound for \( \alpha_i \), so that \( \tilde{\alpha} \leq \alpha_i \leq \text{sup} \alpha_i \) almost surely. Then \( L_i = \mathbb{E}_i [L_i] \). Then for any normalized sampling weight \( w_i = n / p_i \), we have reweighted SGD components [63, Sections 3.2 and 3.3]
\[
\hat{f}_i^{(w)}(\beta) = \frac{1}{w_i} f_i(\beta), \quad F^{(w)}(\beta) = \mathbb{E}^{(w)} [f^{(w)}_i(\beta)], 
\]
\[
L_i^{(w)} = \frac{1}{w_i} L_i = \|x_i\|_2^2 / p_i = \alpha_i, \quad \text{sup} L_i^{(w)} = \text{sup} \frac{1}{w_i} L_i = \hat{L} = \|\tilde{X}\|_F^2 = \text{sup} \alpha_i, \quad \text{and}
\]
\[
\sigma^2_{\text{LS}(w)} = \mathbb{E} \left[ \frac{1}{w_i} \|\nabla f_i(\tilde{X})\|_2^2 \right] = \sum_{i=1}^{n} \frac{\|x_i\|_2^2}{p_i} \|x_i^T \tilde{X} - y_i\|_2^2 \leq \text{sup} \alpha_i \|\tilde{X}\|_F^2 \|y\|_2^2 
\]
and reweighted SGD iterates \( \beta_{k+1} \leftarrow \beta_k - \frac{\gamma}{\tilde{\alpha}} \nabla f_i(\beta_k) \). We rewrite the SGD step-size parameter \( \gamma \) in terms of the step-size parameter \( c \) for the randomized Kaczmarz method in (3.1) with \( \gamma = \frac{c}{\tilde{\alpha}} \). We then update (5.6) with the reweighted SGD components \( L_i^{(w)}, \text{sup} L_i^{(w)}, \) and \( \sigma^2_{\text{LS}(w)} \) with fixed step-size \( c < \frac{\text{sup} \alpha_i}{\text{sup} \alpha_i} \leq 1 \) to obtain
\[
\mathbb{E} \left[ \|\tilde{X}\beta_k - \tilde{X}\beta_*\|_2^2 | \tilde{X} \right] \leq \left( 1 - 2\gamma \mu (1 - \gamma \text{sup} L) \right)^k \mathbb{E} \|\tilde{X}\|_2^2 \|\beta_0 - \beta_*\|_2^2 + \frac{\gamma \sigma^2_{\text{LS}(w)}}{\mu (1 - \gamma \text{sup} L)}, 
\]
(5.7)
where \( r^* = \| \mathbf{X} \hat{\beta} - \mathbf{y} \|_2^2 \). Finally, let \( \hat{\beta}_k \) and \( \hat{\beta}_{LS} \) be the vectors formed from the last \( p \) entries in \( \hat{\beta}_k \) from Algorithm 4.1 and \( \hat{\beta} \), respectively. Since we have an upper bound on the squared Euclidean norm of the \( p+1 \) entries in \( \mathbb{E} \left[ \| \mathbf{X} \hat{\beta}_k - \bar{\mathbf{X}} \hat{\beta} \|_2^2 \mid \mathbf{X} \right] \), the same right hand side is also an upper bound on the \( p \) entries in \( \mathbb{E} \left[ \| \bar{\mathbf{X}} \hat{\beta}_k - \bar{\mathbf{X}} \hat{\beta}_{LS} \|_2^2 \mid \mathbf{X} \right] \).

5.2. Proof of Corollary 4.2. We set the total error to \( \epsilon \cdot (1 + r^*) \) and employ \( c = \frac{\epsilon \alpha}{2(\kappa(\mathbf{X}) \mathbb{E} \| \bar{\mathbf{X}} \|_2^2 + \epsilon \| \mathbf{X} \|_F^2)} \) in the second term on the right of (4.1) to obtain

\[
\frac{c \kappa(\mathbf{X}) \mathbb{E} \| \bar{\mathbf{X}} \|_2^2}{\alpha} \left( 1 - \frac{c}{\alpha} \| \mathbf{X} \|_F^2 \right) \leq \frac{\epsilon}{2}.
\]

Therefore, inserting this \( c \) in the first term on the right of (4.1), requiring that

\[
\left( 1 - 2 \frac{c}{\kappa(\mathbf{X})} \left( 1 - \frac{c}{\alpha} \| \mathbf{X} \|_F^2 \right) \right)^k \mathbb{E} \| \bar{\mathbf{X}} \|_2^2 \epsilon_0 \leq \frac{\epsilon}{2},
\]

and rearranging shows us that we need a \( k \) that satisfies

\[
k \log(1 - A) \leq - \log \left( \frac{2\epsilon_0 \mathbb{E} \| \bar{\mathbf{X}} \|_2^2}{\epsilon} \right),
\]

where \( A = \frac{4\epsilon \mathbb{E} \| \bar{\mathbf{X}} \|_2^2}{\epsilon \| \mathbf{X} \|_F^2 + 2\kappa(\mathbf{X}) \mathbb{E} \| \mathbf{X} \|_F^2} \). Here, we use the fact that \( \inf_i \alpha_i \leq \sup_i \alpha_i = \| \mathbf{X} \|_2^2 \), almost surely. Supposing that \( \mathbf{X} \) and \( \bar{\mathbf{X}} \) are not all zeros matrices, \( A > 0 \). Also, \( A \leq 1 \) for any \( \epsilon < \frac{1}{2} \). Therefore, we can always choose an \( \epsilon \) small enough so that \( 0 < A \leq 1 \). Using the fact that for any \( 0 < A \leq 1 \), \( - \log(1 - A) \leq \frac{1}{A} \) and rearranging gives us

\[
k \geq \log \left( \frac{2\epsilon_0 \mathbb{E} \| \bar{\mathbf{X}} \|_2^2}{\epsilon} \right) \left( \frac{\| \mathbf{X} \|_2^2}{4 \mathbb{E} \| \bar{\mathbf{X}} \|_2^2} + \frac{\kappa(\mathbf{X})}{2\epsilon} \right).
\]

6. Numerical Experiments. We perform numerical experiments with rkLDA from Algorithm 4.1 on real datasets with known labels to investigate several questions. (1) How does rkLDA compare with full-data LDA from (2.4) (gmLDA) and full data least squares LDA from Algorithm 2.1 (lsLDA)? (2) How does performance vary with number of iterations and step-sizes? (3) How does performance vary with smaller and larger values of \( p \)? (4) How does rkLDA compare with randomized matrix sketching (see Section 1.2) and least squares solvers such as CG applied to the normal equations?

To investigate question (3), we perform two sets of experiments on datasets with varying number of features \( p \). (i) The first employs two datasets with a small number of features \( p \) and investigates questions (1) and (2). In particular, we explore how the gmLDA and lsLDA classifiers are aligned, and whether or not the rkLDA iterates can obtain a solution that is close to the gmLDA classifier. (ii) The second set employs two datasets with much larger \( p \) and investigates questions (2) and (4) on larger data, and with comparisons to randomized matrix sketching and CG. Since the solution vectors from larger \( p \) datasets are high-dimensional we do not examine angles in those experiments. Instead, we investigate the change in the LDA discriminant function when employing the RK solution at the \( k^{th} \) iterate in lieu of the full data LS solution \( \| \bar{\mathbf{X}} \hat{\beta}_k - \bar{\mathbf{X}} \hat{\beta}_{LS} \| \). Notably, our results demonstrate that rkLDA can achieve comparable classification accuracy to gmLDA even before the solution vectors are closely aligned.
6.1. Experimental Setup. In the first set of experiments, our comparison metrics include the angle (in degrees) from gmLDA and classification accuracy. We compute the angle without the intercept since gmLDA does not employ one. Whenever possible, we obtain the classifier for gmLDA with the \texttt{lda} function in the \texttt{MASS} package for R since it is the de facto method for performing LDA in R. In cases where the LDA classifier from \texttt{MASS} does not coincide with the gmLDA classifier from (2.4), we instead employ (2.4). For each scenario, we present the mean of the comparison metric over 20 replicates for each combination of step-size $c$ and number of iterations $k$. Values for $c$ are 0.1, 0.3, 0.5, 0.7, and 0.9. Values for $k$ vary by dataset.

In the second set of experiments, our comparison metrics include classification accuracy and time (in seconds). We compute the gmLDA classifier using the \texttt{LinearDiscriminantAnalysis} module from the Scikit-learn package for Python [75]. For each scenario, we present boxplots of the comparison metric over 100 replicates for each combination of step-size $c$ and number of iterations $k$. Values for $c$ are $0.1, 0.3, 0.5, 0.7,$ and $0.9$. Values for $k$ are 500, 1,000, 1,500, 2,000, and 2,500.

The second experiments include comparisons to randomized matrix sketching (MSLDA) and CG. For sketching experiments, we sample the rows of both the design and response for comparability to rkLDA and sample $k$ rows with replacement in each replicate. We do not sample the columns since rkLDA does not perform dimension reduction on the feature space. Additionally, we do not compare other forms of sketching matrices (e.g., Gaussian or FJLT sketching) in order to preserve the values in the response for classification. To obtain the best timing results for sketching, we do not form the sampling matrices and merely subset the selected observations in each replicate. We employ the \texttt{scipy.sparse.linalg.cg} module from the Scipy package for Python [90] to perform CG on the normal equations. We additionally examine the change in the LDA discriminant function when employing the RK solution at the $k^{th}$ iterate in lieu of the full data LS solution $\| \mathbf{X} \hat{\beta}_k - \mathbf{X} \hat{\beta}_{LS} \|$ for varying values of $k$ and step-sizes $c$ over 50 replicates. All experiments were run on an Intel Core i7 3.00 GHz machine with 32 GB of RAM.

6.2. Experiments illustrating angle and accuracy. We present data and results from experiments illustrating angle and accuracy.

6.2.1. Mammographic Mass Data. We employ the mammographic mass dataset [30] containing 830 observations with complete measurements on four features: patient age, mass shape, mass margin, and mass density. The goal is to predict whether a mass is benign or malignant given the observed features. We randomly split the data into training (80 percent) and testing (20 percent) sets and perform 20 replicates of rkLDA on this split so that differences among the replicates are due solely to algorithmic randomness in Algorithm 4.1. The training and testing datasets have rank 4 and scaled condition numbers equal to 7,505 and 7,312, respectively.

Table 2 depicts comparisons between gmLDA, lsLDA, and rkLDA. The top portion reports the coefficients of classifiers from gmLDA, lsLDA, and a single replicate of rkLDA with $c = 0.9$ and $k = 1,000,000$ iterations with the least squares (LS) and optimal intercepts (opt) indicated in subscript. The middle reports the angles of the lsLDA and rkLDA solutions from the gmLDA classifier. The bottom reports overall and class-specific classification accuracy (with classes denoted in subscript) including the intercept whenever possible. Values for the number of iterations $k$ come from an equally spaced sequence of ten numbers from $10^3$ to $10^6.5$. Therefore, $k$ ranges from 1000 to approximately 3,000,000.

The lsLDA and gmLDA coefficients are the same up to a scaling factor of
Therefore, gmLDA and lsLDA produce the same classifiers and the angle between them is 0. Meanwhile, the rkLDA solution after $k = 1,000,000$ iterations is 3.35 degrees from the gmLDA classifier. The lsLDA and rkLDA perform similarly to gmLDA on overall accuracy with some variation in the class-specific accuracy (Accuracy$_1$ and Accuracy$_2$ in Table 2).

Table 2: Top: Coefficients from gmLDA, lsLDA, and rkLDA classifiers with least squares (LS) and optimal (opt) intercepts denoted in subscript. Middle: Angle (degrees) from gmLDA classifier. Bottom: Overall classification accuracy and class-specific accuracy.

$\begin{array}{cccccccc}
\text{gmLDA} & \text{lsLDA}_{\text{LS}} & \text{lsLDA}_{\text{opt}} & \text{Scaling} & \text{rkLDA}_{\text{LS}} & \text{rkLDA}_{\text{opt}} & \text{Scaling} \\
\text{Intercept} & -3.17 & -3.17 & -3.05 & -3.49 & -3.05 & -3.49 \\
\text{Age} & 0.02 & 0.02 & 0.02 & 0.99 & 0.03 & 0.03 & 0.93 \\
\text{Shape} & 0.50 & 0.51 & 0.51 & 0.99 & 0.54 & 0.54 & 0.94 \\
\text{Margin} & 0.40 & 0.40 & 0.40 & 0.99 & 0.40 & 0.40 & 1.00 \\
\text{Density} & -0.20 & -0.20 & -0.20 & 0.99 & -0.25 & -0.25 & 0.82 \\
\text{Angle} & 0.00 & & & & & 3.35 & \\
\text{Accuracy} & 0.80 & 0.79 & 0.79 & 0.79 & 0.80 & 0.80 & \\
\text{Accuracy}_1 & 0.74 & 0.71 & 0.75 & 0.69 & 0.76 & 0.76 & \\
\text{Accuracy}_2 & 0.85 & 0.88 & 0.84 & 0.88 & 0.88 & 0.84 & \\
\end{array}$

Figure 1a depicts the mean angle between the gmLDA and rkLDA classifiers obtained over 20 replicates as a function of the number of iterations $k$. Colored lines depict results from different step-size values $c$. While results with all step-sizes appear to approach the gmLDA classifier, larger step-sizes converge more quickly to a solution that is very close to the gmLDA classifier. Figure 1b depicts the mean overall classification accuracy of the rkLDA classifier obtained over 20 replicates with the optimal intercept as a function of the number of iterations $k$. The dashed gray line depicts the gmLDA overall classification accuracy. Notably, accuracy from all choices of step-size $c$ approach the gmLDA accuracy even before the angle from the gmLDA classifier becomes very small.

![Mean angle (degrees) between LDA and rkLDA classifiers.](image1)
![Mean rkLDA accuracy. Gray line depicts gmLDA accuracy (0.797).](image2)

Fig. 1: Mammographic mass results over 20 replicates as a function of iterations.

### 6.2.2. Occupancy Detection Data

We employ the occupancy detection dataset from [11] containing 8,143 training and 9,752 testing observations on four features: temperature (in Celsius), relative humidity (percent), light (in lux), and CO$_2$ (in ppm). The goal is to predict whether or not a room is occupied given the observed features. We perform 20 replicates of rkLDA on the training data so that...
differences among the replicates are due solely to algorithmic randomness from the sampling procedure in Algorithm 4.1. The training and testing datasets have rank 4 and scaled condition numbers equal to 43,979 and 120,589, respectively.

Table 3 depicts comparisons between gmLDA, lsLDA, and rkLDA. The top portion reports the coefficients of the classifiers from gmLDA, lsLDA, and a single iterate of rkLDA with $c = 0.9$ and $k = 100,000$ iterations with the least squares (LS) and optimal intercepts (opt) indicated in subscript. The middle reports the angles of the lsLDA and rkLDS solutions from the gmLDA classifier. The bottom reports the overall and class-specific classification accuracy (with classes denoted in subscript) including the intercept whenever possible. Values for the number of iterations $k$ come from an equally spaced sequence of ten numbers from $10^3$ to $10^6$. Therefore, $k$ ranges from 1000 to approximately 1,000,000.

|        | gmLDA | lsLDA$_{LS}$ | lsLDA$_{opt}$ | Scaling | rkLDA$_{LS}$ | rkLDA$_{opt}$ | Scaling |
|--------|-------|--------------|---------------|---------|--------------|--------------|---------|
| Intercept | 5.65  | 2.86         | 0.94          | -2.28   | -2.28        | -2.28        |         |
| Temperature | -0.44 | -0.38        | -0.38         | 1.17    | -0.14        | -0.14        | 3.17    |
| Humidity  | -0.02 | -0.01        | -0.01         | 1.17    | -0.01        | -0.01        | 1.35    |
| Light     | 0.01  | 0.01         | 0.01          | 1.17    | 0.01         | 0.01         | 0.99    |
| CO$_2$    | 0.00  | 0.00         | 0.00          | 1.17    | 0.00         | 0.00         | 1.78    |
| Angle     |       |              | 4.63          |         |              |              |         |
| Accuracy  | 0.99  | 0.88         | 0.98          | 0.94    | 0.99         | 0.99         |         |
| Accuracy$_1$ | 0.99  | 0.85         | 0.99          | 0.93    | 0.99         | 0.99         |         |
| Accuracy$_2$ | 1.00  | 1.00         | 0.93          | 1.00    | 0.99         | 0.99         |         |

Table 3: Top: Coefficients from gmLDA, lsLDA, and rkLDA classifiers with least squares (LS) and optimal (opt) intercepts denoted in subscript. Middle: Angle (degrees) from gmLDA classifier. Bottom: Overall classification accuracy and class-specific accuracy.

The lsLDA and gmLDA coefficients are the same up to a scaling factor of 1.17. Therefore, gmLDA and lsLDA produce the same classifiers and the angle between them is 0. Meanwhile, the rkLDA solution after $k = 100,000$ iterations is 4.63 degrees from the gmLDA classifier. Both lsLDA and rkLDA produce overall accuracy that is very comparable to that of gmLDA.

Figure 2a depicts mean angle (in degrees) between gmLDA and rkLDA vectors from 20 replicates as a function of iterations $k$. Colored lines depict results from different step-sizes $c$. Results with all step-sizes appear to approach the gmLDA classifier and larger step-sizes converge more quickly to a solution that is close to the gmLDA classifier. Figure 2b depicts mean overall classification accuracy of the rkLDA solution from 20 replicates with the optimal intercept as a function of iterations $k$. Colored lines depict results from different step-sizes $c$ while the dashed gray line depicts gmLDA accuracy. Again, accuracy with all step-sizes $c$ approach gmLDA accuracy even before the angle from the gmLDA classifier is small.

6.3. Larger $p$ experiments illustrating accuracy and timing. We present data and results from larger $p$ experiments illustrating accuracy and timing with comparisons to matrix sketching (Section 1.2) and CG.

6.3.1. Modified National Institute of Standards and Technology (MNIST) Dataset. We employ the MNIST dataset from [21] containing 70,000 images with $p = 784$ pixels $(28 \times 28)$ each. To identify similar digits for a more representative comparison, we perform PCA on all the MNIST digits and plot their representations in the first and second principal components. Observing that the
digits 6 and 8 exhibit substantial overlapping points in the first and second principal components, we subset the data for these two digits, resulting in $n = 13,701$ images. The training and testing datasets have scaled condition numbers $1.1 \times 10^{11}$ and $1.0 \times 10^{11}$, respectively. This is largely due to the training and test datasets having squared Frobenius norm equal to $3.5 \times 10^{10}$ and $8.8 \times 10^{9}$, respectively. In this data, 78 percent of the entries are zero.

![Image](image_url)

(a) Mean angle (degrees) between LDA and rkLDA classifiers. (b) Mean rkLDA accuracy. Gray line depicts gmLDA accuracy (0.991).

Fig. 2: Occupancy detection results over 20 replicates as a function of iterations.

| Method   | $k=500$ | $k=1000$ | $k=1,500$ | $k=2,000$ | $k=2,500$ |
|----------|---------|----------|-----------|-----------|-----------|
| Full data gmLDA | 0.9836  | –        | –         | –         | –         |
| Full data CG   | 0.9839  | –        | –         | –         | –         |

**Uniform weights**

| rkLDA c=0.1 | 0.9787 (0.0021) | 0.9806 (0.0015) | 0.9813 (0.0011) | 0.9819 (0.0014) | 0.9827 (0.0014) |
| rkLDA c=0.3 | **0.9800** (0.0025) | **0.9816** (0.0019) | **0.9826** (0.0018) | **0.9827** (0.0017) | **0.9829** (0.0015) |
| rkLDA c=0.5 | 0.9790 (0.0031) | 0.9811 (0.0023) | 0.9810 (0.0021) | 0.9820 (0.0024) | 0.9817 (0.0019) |
| rkLDA c=0.7 | 0.9771 (0.0035) | 0.9793 (0.0035) | 0.9799 (0.0027) | 0.9793 (0.0038) | 0.9799 (0.0025) |
| rkLDA c=0.9 | 0.9744 (0.0045) | 0.9770 (0.0034) | 0.9772 (0.0038) | 0.9766 (0.0052) | 0.9770 (0.0035) |
| MSLDA      | 0.6682 (0.0438) | 0.9318 (0.0069) | 0.9558 (0.0043) | 0.9665 (0.0038) | 0.9708 (0.0033) |

**Row-based weights**

| rkLDA c=0.1 | 0.9777 (0.0018) | 0.9803 (0.0016) | 0.9812 (0.0022) | **0.9831** (0.0012) | 0.9826 (0.0006) |
| rkLDA c=0.3 | **0.9809** (0.0020) | **0.9821** (0.0011) | **0.9831** (0.0014) | 0.9821 (0.0008) | **0.9837** (0.0012) |
| rkLDA c=0.5 | 0.9779 (0.0024) | 0.9788 (0.0018) | 0.9815 (0.0017) | 0.9819 (0.0012) | 0.9828 (0.0013) |
| rkLDA c=0.7 | 0.9756 (0.0028) | 0.9796 (0.0028) | 0.9815 (0.0003) | 0.9799 (0.0033) | 0.9802 (0.0039) |
| rkLDA c=0.9 | 0.9737 (0.0021) | 0.9792 (0.0022) | 0.9765 (0.0024) | 0.9752 (0.0034) | 0.9776 (0.0021) |
| MSLDA      | 0.6731 (0.0461) | 0.9315 (0.0071) | 0.9561 (0.0045) | 0.9655 (0.0037) | 0.9704 (0.0034) |

Table 4: MNIST Accuracy Results – Mean classification accuracy (and standard deviations) over varying values of $k$ (RK iterates for rkLDA and sketched dimension for MSLDA) and 100 replicates. Full data LDA and conjugate gradient (CG) results do not utilize $k$ and produce same results in all replicates. Most accurate mean results bolded in each of the uniform and row-based weights blocks.

Tables 4 and 5 depict mean classification accuracy and time results (with standard deviations in italics and parentheses) from full data gmLDA, full data CG, rkLDA, and MSLDA. The gmLDA and CG solutions are non-random and therefore unchanged over replicates. The rkLDA and MSLDA results are over 100 replicates for each value of $k$ (RK iterates for rkLDA and sketched dimension for MSLDA). The middle section employs uniform sampling weights while the bottom employs row-based sampling weights, where the $i^{th}$ row was sampled with probability $p_i = \frac{\|s_i\|^2}{\|X\|^2}$.

Accuracy results show that rkLDA can obtain accuracy comparable to gmLDA even with a small number of iterations ($500 \leq k \leq 2,500$) and even when the step-size
Table 5: MNIST Timing Results – Mean classification time in seconds (and standard deviations) over varying values of $k$ (RK iterates for rkLDA and sketched dimension for MSLDA) and 100 replicates. Full data LDA and conjugate gradient (CG) over 5 replicates do not utilize $k$. Fastest mean results bolded in each of the uniform and row-based weights blocks.

| Method | $k=500$  | $k=1,000$ | $k=1,500$ | $k=2,000$ | $k=2,500$ |
|--------|----------|-----------|-----------|-----------|-----------|
| gmLDA  | 0.9388 (0.0744) | – | – | – | – |
| CG     | 1.8179 (0.0583) | – | – | – | – |

Uniform weights

| rkLDA | $c=0.1$ | 0.0985 (0.0009) | 0.1077 (0.0036) | **0.1131** (0.0014) | **0.1246** (0.0074) | 0.1296 (0.0102) |
|--------|----------|----------------|-----------------|---------------------|---------------------|------------------|
| rkLDA  | $c=0.3$ | 0.0986 (0.0008) | 0.1078 (0.0043) | 0.1132 (0.0010) | 0.1249 (0.0077) | 0.1296 (0.0104) |
| rkLDA  | $c=0.5$ | 0.0986 (0.0009) | 0.1074 (0.0026) | **0.1131** (0.0009) | 0.1249 (0.0076) | 0.1298 (0.0114) |
| rkLDA  | $c=0.7$ | 0.0986 (0.0021) | **0.1073** (0.0023) | **0.1131** (0.0009) | 0.1251 (0.0079) | 0.1297 (0.0107) |
| rkLDA  | $c=0.9$ | 0.0984 (0.0007) | 0.1075 (0.0022) | **0.1131** (0.0012) | 0.1249 (0.0076) | **0.1294** (0.0102) |
| MSLDA  | **0.0601** (0.0033) | 0.1228 (0.0055) | 0.1697 (0.0029) | 0.2189 (0.0025) | 0.2583 (0.0027) |

Row-based weights

| rkLDA | $c=0.1$ | 0.1034 (0.0011) | 0.1112 (0.0005) | 0.1223 (0.0011) | 0.1356 (0.0015) | 0.1424 (0.0015) |
|--------|----------|----------------|-----------------|-----------------|----------------|------------------|
| rkLDA  | $c=0.3$ | 0.1039 (0.0008) | 0.1112 (0.0012) | 0.1217 (0.0011) | 0.1372 (0.0012) | 0.1423 (0.0010) |
| rkLDA  | $c=0.5$ | 0.1086 (0.0096) | 0.1132 (0.0044) | **0.1212** (0.0009) | 0.1352 (0.0019) | **0.1417** (0.0007) |
| rkLDA  | $c=0.7$ | 0.1041 (0.0008) | **0.1111** (0.0009) | 0.1217 (0.0011) | **0.1351** (0.0009) | 0.1421 (0.0009) |
| rkLDA  | $c=0.9$ | 0.1044 (0.0004) | **0.1111** (0.0005) | 0.1215 (0.0005) | 0.1355 (0.0005) | 0.1430 (0.0012) |
| MSLDA  | **0.0604** (0.0056) | 0.1237 (0.0067) | 0.1719 (0.0076) | 0.2201 (0.0095) | 0.2606 (0.0110) |

is 0.9, or close to 1. Timing results show that rkLDA achieves these results about 10 times faster than gmLDA on this dataset. MSLDA is faster than rkLDA at smaller $k$ but suffers worse accuracy. As $k$ increases, sketching accuracy drastically improves but also requires more time. On this dataset, sketching is generally slower than rkLDA with all step-sizes, and less accurate at even the largest sketching dimension and for both uniform and row-based sampling weights. Finally, CG has similar accuracy as gmLDA on this dataset but was substantially slower than rkLDA and matrix sketching at all values of $k$.

Figure 3 depicts boxplots of the classification accuracy at the $k$th iterate for varying values of $k$ and step-sizes $c$ over 50 replicates. Each set of 5 boxplots over each value of $k$ indicates results for rkLDA with $c = 0.1, 0.3, 0.5, 0.7$ and 0.9 (from left to right).
4 shows that this was a very good problem for full data LDA and Figure 3 shows visually how similarly high accuracy results can be obtained with a relatively small number of RK iterates $k$. Note that the values are not identical to those in Table 4 since the results are from two different sets of experiments.

**6.3.2. CIFAR-10 Dataset.** We employ the CIFAR-10 dataset from [48] containing 50,000 training and 10,000 test images with $p = 3,072$ pixels (1,024 entries for each of the three RGB channels). To identify similar classes for a more representative comparison, we perform PCA on all the CIFAR-10 training images and plot their representations in the first and second principal components. Observing that the groups containing ships and trucks exhibit substantial overlapping points in the first and second principal components, we subset the data for these two groups, resulting in 10,000 training images and 2,000 test images, equally split between the two classes. The training and testing datasets have scaled condition numbers $3.3 \times 10^{11}$ and $5.3 \times 10^{10}$, respectively. This is largely due the training and test datasets having squared Frobenius norm equal to $6.5 \times 10^{11}$ and $1.3 \times 10^{11}$, respectively. Compared to MNIST, this dataset is dense; less than 0.1 percent of the entries are zero. Since $k < p$ at all choices of $k$, we employ lsqr, rather than the default solver, in the Scikit-learn LinearDiscriminantAnalysis module to perform gmLDA after matrix sketching.

| Method         | k=500 | k=1,000 | k=1,500 | k=2,000 | k=2,500 |
|----------------|-------|---------|---------|---------|---------|
| Full data gmLDA | 0.7280 | –       | –       | –       | –       |
| Full data CG    | 0.6670 | –       | –       | –       | –       |
| Uniform weights |       |         |         |         |         |
| rkLDA c=0.1     | 0.6618 (0.0623) | 0.7021 (0.0573) | **0.7125** (0.0327) | **0.7264** (0.0276) | **0.7326** (0.0223) |
| rkLDA c=0.3     | 0.6761 (0.0508) | **0.7049** (0.0481) | 0.7093 (0.0529) | 0.7263 (0.0416) | 0.7241 (0.0432) |
| rkLDA c=0.5     | 0.6745 (0.0626) | 0.6900 (0.0615) | 0.6975 (0.0625) | 0.6992 (0.0590) | 0.6969 (0.0710) |
| rkLDA c=0.7     | 0.6461 (0.0700) | 0.6726 (0.0719) | 0.6720 (0.0753) | 0.6720 (0.0714) | 0.6901 (0.0721) |
| rkLDA c=0.9     | 0.6467 (0.0722) | 0.6547 (0.0704) | 0.6449 (0.0689) | 0.6628 (0.0711) | 0.6778 (0.0677) |
| MSLDA           | 0.4980 (0.0361) | 0.4980 (0.0305) | 0.4985 (0.0188) | 0.5007 (0.0167) | 0.5011 (0.0162) |
| Row-based weights |     |         |         |         |         |
| rkLDA c=0.1     | 0.6900 (0.0510) | **0.7030** (0.0361) | **0.7238** (0.0263) | **0.7261** (0.0240) | **0.7311** (0.0253) |
| rkLDA c=0.3     | 0.6797 (0.0577) | 0.7015 (0.0517) | 0.7081 (0.0467) | 0.7228 (0.0442) | 0.7223 (0.0462) |
| rkLDA c=0.5     | 0.6783 (0.0634) | 0.6828 (0.0616) | 0.6964 (0.0589) | 0.7127 (0.0516) | 0.6973 (0.0628) |
| rkLDA c=0.7     | 0.6466 (0.0665) | 0.6607 (0.0749) | 0.6755 (0.0687) | 0.6772 (0.0742) | 0.6753 (0.0619) |
| rkLDA c=0.9     | 0.6321 (0.0779) | 0.6653 (0.0727) | 0.6610 (0.0712) | 0.6630 (0.0687) | 0.6607 (0.0773) |
| MSLDA           | 0.5047 (0.0277) | 0.4998 (0.0211) | 0.4980 (0.0155) | 0.5009 (0.0158) | 0.4974 (0.0184) |

Table 6: CIFAR Accuracy Results – Mean classification accuracy (and standard deviations) over varying values of $k$ (RK iterates for rkLDA and sketched dimension for MSLDA) and 100 replicates. Full data LDA and conjugate gradient (CG) results do not utilize $k$ and produce same results in all replicates. Most accurate mean results bolded in each of the uniform and row-based weights blocks.

Tables 6 and 7 depict mean classification accuracy and time results (with standard deviations in italics and parentheses) from full data gmLDA, full data CG, rkLDA, and MSLDA. Accuracy results show that rkLDA can obtain accuracy comparable to gmLDA even with a small number of iterations ($2,000 \leq k \leq 2,500$). In some cases, rkLDA can even outperform gmLDA, as the mean accuracy in the $c = 0.1$ case exceeds gmLDA for $k \geq 2,500$. Timing results show that rkLDA is about 10 times faster than gmLDA on this dataset. MSLDA with the same $k$ as in rkLDA does not perform well here since $k < p$ in all cases and neither random row sampling nor gmLDA are intended for underdetermined systems. CG with default tolerance for stopping condition of $10^{-5}$ is less accurate than both gmLDA and mean rkLDA for some choices of $c$ at all $k$ on this dataset. Decreasing the tolerance increases runtime but does not improve accuracy on this dataset; CG does not converge at a tolerance
Table 7: CIFAR Timing Results – Mean classification time in seconds (and standard deviations) over varying values of $k$ (RK iterates for rkLDA and sketched dimension for MSLDA) and 100 replicates. Full data LDA and conjugate gradient (CG) over 5 replicates do not utilize $k$. Fastest mean results bolded in each of the uniform and row-based weights blocks.

| Method       | $k=500$       | $k=1,000$     | $k=1,500$     | $k=2,000$     | $k=2,500$     |
|--------------|--------------|--------------|--------------|--------------|--------------|
| Full data gmLDA | 11.5235 ($0.1246$) | –            | –            | –            | –            |
| Full data CG   | 116.2896 ($2.8152$) | –            | –            | –            | –            |
| Uniform weights|              |              |              |              |              |
| rkLDA $c=0.1$  | 0.8090 ($0.0215$) | 0.8305 ($0.0065$) | **0.8489** ($0.0142$) | 0.8585 ($0.0146$) | 0.8670 ($0.0151$) |
| rkLDA $c=0.3$  | 0.8093 ($0.0222$) | 0.8110 ($0.0084$) | 0.8505 ($0.0234$) | **0.8561** ($0.0069$) | 0.8699 ($0.0196$) |
| rkLDA $c=0.5$  | 0.8091 ($0.0213$) | 0.8307 ($0.0068$) | 0.8573 ($0.0561$) | 0.8604 ($0.0257$) | 0.8723 ($0.0457$) |
| rkLDA $c=0.7$  | 0.8144 ($0.0514$) | **0.8300** ($0.0041$) | 0.8533 ($0.0580$) | 0.8566 ($0.0099$) | **0.8659** ($0.0112$) |
| rkLDA $c=0.9$  | **0.8090** ($0.0198$) | 0.8303 ($0.0084$) | 0.8541 ($0.0331$) | 0.8611 ($0.0300$) | 0.8697 ($0.0226$) |
| MSLDA         | 3.8442 ($0.0825$) | 3.9919 ($0.0866$) | 4.1563 ($0.0969$) | 4.3069 ($0.1077$) | 4.4227 ($0.1022$) |
| Row-based weights|             |              |              |              |              |
| rkLDA $c=0.1$  | 0.8352 ($0.0235$) | **0.8467** ($0.0062$) | 0.8652 ($0.0240$) | **0.8934** ($0.0111$) | 0.9144 ($0.0171$) |
| rkLDA $c=0.3$  | **0.8311** ($0.0050$) | 0.8470 ($0.0140$) | 0.8686 ($0.0364$) | 0.8965 ($0.0254$) | **0.9126** ($0.0079$) |
| rkLDA $c=0.5$  | 0.8383 ($0.0186$) | 0.8496 ($0.0192$) | 0.8618 ($0.0068$) | 0.8979 ($0.0494$) | 0.9129 ($0.0114$) |
| rkLDA $c=0.7$  | 0.8371 ($0.0193$) | 0.8483 ($0.0182$) | 0.8634 ($0.0166$) | 0.8941 ($0.0126$) | 0.9130 ($0.0129$) |
| rkLDA $c=0.9$  | 0.8372 ($0.0483$) | **0.8467** ($0.0065$) | **0.8615** ($0.0092$) | 0.8978 ($0.0342$) | 0.9137 ($0.0107$) |
| MSLDA         | 3.8925 ($0.0564$) | 4.0425 ($0.0501$) | 4.2091 ($0.0638$) | 4.3733 ($0.1116$) | 4.4724 ($0.0678$) |

Fig. 4: CIFAR-10 Classification Accuracy – Boxplots of the classification accuracy for varying RK iterates $k$ and step-sizes $c$ over 50 replicates. Each set of 5 boxplots over each value of $k$ indicates results for rkLDA with $c = 0.1, 0.3, 0.5, 0.7$ and 0.9 (from left to right).

Figure 4 depicts boxplots of the classification accuracy at the $k^{th}$ iterate for varying values of $k$ and step-sizes $c$ over 50 replicates and row-based weights. Table 6 shows that this was a difficult problem for full data LDA and Figure 4 shows visually how similar accuracy results can be obtained with a relatively small number of RK iterates $k$. Note that the values are not identical to those in Table 6 since the results are from two different sets of experiments.

7. Discussion. In this work, we present rkLDA, an iterative randomized approach to binary-class Gaussian model LDA for very large data. The keys to our
method and analysis lie in harnessing the least squares formulation of binary-class Gaussian model LDA, employing the machinery of the stochastic gradient descent framework with importance sampling [63], and combining these with previous work on the statistical analysis of sketched linear regression [15, 56]. Therefore, our analysis of the expected deviation of the LDA discriminant function on new data when employing the RK solution in lieu of the least squares one after a finite number of steps account for both randomness from the data modeling assumptions and algorithmic randomness from the RK iterates.

Although the RK is a very potent tool for solving large linear systems of equations, it does have some limitations. First, the RK convergence rate depends on the conditioning of $X$ [84, Theorem 2] so that RK iterates from ill-conditioned $X$ converge more slowly. However, our numerical experiments demonstrate that a relatively small number of iterations may be sufficient to obtain classification accuracy that is very comparable to that of LDA – even when the scaled condition number of the training data is relatively high. Therefore, it is possible to obtain a sufficiently accurate $\hat{\beta}$ for classification purposes with a relatively small number of iterates. Nonetheless, one possible direction of future work for enhancing the rkLDA convergence rate for ill-conditioned design matrices may be to employ an acceleration method such as randomized block Kaczmarz [64] – which can work well if the rows of $X$ can be partitioned into well-conditioned blocks, or selectable set RK [97] – which leverages information about the iterates to obtain a suitable subset of equations to be sampled from in each iteration. Additionally, the RK iterates are susceptible to noise. In particular, the least squares formulation of LDA in (2.6) assumes noise in the response since observations are class-conditionally Gaussian. Therefore, the convergence horizon in Theorem 4.1 depends on both the step-size $c$ and the size of the least squares residuals $r^\star$.

We comment briefly on the convergence of the RK to a horizon of the least squares solution. Empirically, our numerical experiments show that in the context of classification, an approximate solution that is within a horizon of the least squares solution can perform quite well. Moreover, it is possible to achieve such a sufficiently accurate solution with a relatively small number of iterates and without exceedingly small values for the step-size parameter $c$. More sophisticated variations such as SGD with decreasing step-size or the randomized extended Kaczmarz [103] are able to converge to the least squares solution. To keep our contributions in applying an RK-type method to LDA clear, we focus on analyzing the RK without the need to include non-obvious decreasing step-size agendas or extension methods that require column-wise computations. Nonetheless, it could be interesting future work to see whether or not there are additional gains in the classification context to be had from achieving a more accurate approximate solution.

While we consider the overdetermined $n > p$ case in this work, many applications such as image recognition and text analysis can involve $p > n$. This ultrahigh dimension case is a well-known problem in Gaussian model LDA since $\hat{\Sigma}^{-1}$ in (2.4) does not exist in this case [57] and there have been a number of works addressing this issue [5, 71, 51, 68]. Therefore, another avenue of future work may be to adapt the analysis and optimal intercept in (2.8) for ultrahigh dimensions. Since (2.8) involves computing $\hat{\Sigma} \in \mathbb{R}^{p \times p}$, this may become computationally prohibitive for $p$ sufficiently large. One possibility may be to employ randomized approaches to computing the sample covariance. This would involve exploring how these approximations in the optimal intercept may affect classification accuracy.
Finally, we briefly comment on how one can employ rkLDA when the data may be too large to store in working memory on a local machine. Since Algorithm 4.1 only requires one row of the training data in each iteration, one may store and compute iterates on batches of sampled rows from the training data sequentially. In this way, one bypasses the need to store the full data. We note that unlike the leverage scores, the row norms and $\Sigma$ for the optimal intercept in (2.8) can also be computed by viewing only batches of the data sequentially. The idea of viewing batches of the data in sequence is similar to streaming discriminant analysis [73, 40, 3]. While these are viable alternatives to full data analysis in the streaming cases, a primary difference is that these lack theoretical convergence guarantees for how close the resulting classifier is to the original full data classifier. Additionally, our numerical experiments demonstrate that rkLDA can be a very viable alternative to LDA in terms of both accuracy and timing when the entire training and test data fit in local memory. Therefore, another promising line of future work would be to employ parallel RK algorithms [45, 46, 55, 92] to adapt rkLDA for distributed computing.

REFERENCES

[1] A. Agaskar, C. Wang, and Y. M. Lu, Randomized Kaczmarz algorithms: Exact MSE analysis and optimal sampling probabilities, in 2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP), IEEE, 2014, pp. 389–393.

[2] N. Ailon and B. Chazelle, Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform, in Proceedings of the 38th Annual ACM Symposium on Theory of Computing, 2006, pp. 557–563.

[3] C. Anagnostopoulos, D. K. Tasoulis, N. M. Adams, N. G. Pavlidis, and D. J. Hand, Online linear and quadratic discriminant analysis with adaptive forgetting for streaming classification, Statistical Analysis and Data Mining: The ASA Data Science Journal, 5 (2012), pp. 139–166.

[4] H. Avron, P. Maymounkov, and S. Toledo, Blendenpik: Supercharging lapack’s least-squares solver, SIAM Journal on Scientific Computing, 32 (2010), pp. 1217–1236.

[5] P. J. Bickel, E. Levina, et al., Some theory for Fisher’s linear discriminant function, naive Bayes’, and some alternatives when there are many more variables than observations, Bernoulli, 10 (2004), pp. 989–1010.

[6] L. Bottou, Large-scale machine learning with stochastic gradient descent, in Proceedings of COMPSTAT’2010, Springer, 2010, pp. 177–186.

[7] L. Bottou, Stochastic gradient descent tricks, in Neural Networks: Tricks of the Trade, Springer, 2012, pp. 421–436.

[8] C. Boutsidis and P. Drineas, Random projections for the nonnegative least-squares problem, Linear Algebra and its Applications, 431 (2009), pp. 760–771.

[9] L. Breiman and R. Iha, Nonlinear discriminant analysis via scaling and ace, tech. report, Department of Statistics, University of California Davis, 1984.

[10] Q. Cai, J. W. Salmon, and M. E. Rodgers, Factors associated with long-stay nursing home admissions among the US elderly population: comparison of logistic regression and the cox proportional hazards model with policy implications for social work, Social work in health care, 48 (2009), pp. 154–168.

[11] L. M. Candenego and V. Feldheim, Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models, Energy and Buildings, 112 (2016), pp. 28–39.

[12] Y. Censor, P. P. Eggermont, and D. Gordon, Strong underrelaxation in Kaczmarz’s method for inconsistent systems, Numerische Mathematik, 41 (1983), pp. 83–92.

[13] Y. Censor, G. T. Herman, and M. Jiang, A note on the behavior of the randomized Kaczmarz algorithm of Strohmer and Vershynin, Journal of Fourier Analysis and Applications, 15 (2009), pp. 431–436.

[14] J. T. Chi and I. C. F. Ipsen, Multiplicative perturbation bounds for multivariate multiple linear regression in Schatten $p$-norms, Linear Algebra and its Applications, 624 (2021), pp. 87–102.

[15] J. T. Chi and I. C. F. Ipsen, A projector-based approach to quantifying total and excess uncertainties for sketched linear regression, Information and Inference, 11 (2022),
[16] A. Chowdhury, J. Yang, and P. Drineas, Randomized iterative algorithms for fisher discriminant analysis, in Uncertainty in Artificial Intelligence, PMLR, 2020, pp. 239–249.

[17] T. Cooke, Two variations on Fisher’s linear discriminant for pattern recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24 (2002), pp. 268–273.

[18] L. Dai, M. Soltanalian, and K. Pelckmans, On the randomized Kaczmarz algorithm, IEEE Signal Processing Letters, 21 (2013), pp. 330–333.

[19] K. R. Davidson and S. J. Szarek, Local operator theory, random matrices and banach spaces, Handbook of the geometry of Banach spaces, 1 (2001), p. 131.

[20] J. W. Demmel, The probability that a numerical analysis problem is difficult, Mathematics of Computation, 50 (1988), pp. 449–480.

[21] L. Deng, The mnist database of handwritten digit images for machine learning research [best of the web], IEEE signal processing magazine, 29 (2012), pp. 141–142.

[22] R. J. Desai, S. V. Wang, M. Vaduganathan, T. Evers, and S. Schneeweiss, Comparison of Machine Learning Methods With Traditional Models for Use of Administrative Claims With Electronic Medical Records to Predict Heart Failure Outcomes, JAMA Network Open, 3 (2020), pp. e1918962–e1918962, https://doi.org/10.1001/jamanetworkopen.2019.18962.

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

[24] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, Sampling algorithms for l_2 regression and applications, in Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, 2006, pp. 1127–1136.

[25] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós, Faster least squares approximation, Numerische Mathematik, 117 (2011), pp. 219–249.

[26] Q. Du, Modified Fisher’s linear discriminant analysis for hyperspectral imagery, IEEE Geoscience and Remote Sensing letters, 4 (2007), pp. 503–507.

[27] R. J. Durrant and A. Kabán, Compressed Fisher linear discriminant analysis: Classification of randomly projected data, in Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2010, pp. 1119–1128.

[28] R. J. Durrant and A. Kabán, Random projections as regularizers: Learning a linear discriminant from fewer observations than dimensions, Journal of Machine Learning Research, 99 (2015), pp. 257–286.

[29] A. Edelman, Eigenvalues and Condition Numbers of Random Matrices, PhD thesis, Yale University, 1989.

[30] M. Elter, R. Schulz-Wendtland, and T. Wittenberg, The prediction of breast cancer biopsy outcomes using two cad approaches that both emphasize an intelligible decision process, Medical Physics, 34 (2007), pp. 4164–4172.

[31] A. Filatova, V. Vybornov, E. Pylaeva, A. Potekhina, T. Arefieva, and S. Provatorov, Machine learning in prediction of in-stent restenosis, Atherosclerosis, 355 (2022), p. 259, https://doi.org/10.1016/j.atherosclerosis.2022.06.982, https://www.sciencedirect.com/science/article/pii/S0021915022012692.

[32] R. A. Fisher, The use of multiple measurements in taxonomic problems, Annals of Eugenics, 7 (1936), pp. 179–188.

[33] T. Fitzpatrick and C. Mues, An empirical comparison of classification algorithms for mortgage default prediction: evidence from a distressed mortgage market, European Journal of Operational Research, 249 (2016), pp. 427–439.

[34] G. H. Golub and C. F. Van Loan, Matrix Computations, The Johns Hopkins University Press, Baltimore, fourth ed., 2013.

[35] F. Guimet, R. Boqué, and J. Ferré, Application of non-negative matrix factorization combined with Fisher’s linear discriminant analysis for classification of olive oil excitation–emission fluorescence spectra, Chemometrics and Intelligent Laboratory Systems, 81 (2006), pp. 94–106.

[36] M. Hanke, Conjugate gradient type methods for ill-posed problems, vol. 327, CRC Press, 1995.

[37] M. Hanke and W. Niethammer, On the acceleration of kaczmarz’s method for inconsistent linear systems, Linear Algebra and its Applications, 130 (1990), pp. 83–98.

[38] T. Hastie, R. Tibshirani, and J. Friedman, The elements of statistical learning: data mining, inference, and prediction, Springer Science & Business Media, 2009.

[39] R. P. Hauser and D. Booth, Predicting bankruptcy with robust logistic regression, Journal of Data Science, 9 (2011), pp. 565–584.

[40] T. L. Hayes and C. Kanan, Lifelong machine learning with deep streaming linear
discriminant analysis, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 220–221.

[41] I. C. IPSEN AND T. WENTWORTH, The effect of coherence on sampling from matrices with orthonormal columns, and preconditioned least squares problems, SIAM Journal on Matrix Analysis and Applications, 35 (2014), pp. 1490–1520.

[42] A. KARÅN, New bounds on compressive linear least squares regression, in Artificial Intelligence and Statistics, PMLR, 2014, pp. 448–456.

[43] S. KACZMARZ, Angenaherte auflosung von systemen linearer gleichungen: Bulletin international de l'académie polonaise des sciences et des lettres, Classe des Sciences Mathématiques et Naturelles. Série A, Sciences Mathématiques, 35 (1937), pp. 355–357.

[44] L. KAISER, Why do low-income women not use food stamps? findings from the california women’s health survey, Public health nutrition, 11 (2008), pp. 1288–1295.

[45] G. KAMATH, P. RAMANAN, AND W.-Z. SONG, Distributed randomized Kaczmarz and applications to seismic imaging in sensor network, in 2015 International Conference on Distributed Computing in Sensor Systems, IEEE, 2015, pp. 169–178.

[46] F. KEINERT AND E. S. WEBER, A randomized distributed Kaczmarz algorithm and anomaly detection, Axioms, 11 (2022), p. 106.

[47] H. KIM AND Z. GU, A logistic regression analysis for predicting bankruptcy in the hospitality industry, The Journal of Hospitality Financial Management, 14 (2006), pp. 17–34.

[48] A. KRIZHEVSKY, G. HINTON, ET AL., Learning multiple layers of features from tiny images. https://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf, 2009.

[49] E. K. LAITINEN AND T. LAITINEN, Bankruptcy prediction: Application of the taylor’s expansion in logistic regression, International review of financial analysis, 9 (2000), pp. 327–349.

[50] A. F. LAPANOWSKI AND I. GAYNANOVA, Compressing large sample data for discriminant analysis, in 2021 IEEE International Conference on Big Data (Big Data), IEEE, 2021, pp. 1068–1076.

[51] Y. LI, H. G. HONG, AND Y. LI, Multiclass linear discriminant analysis with ultrahigh-dimensional features, Biometrics, 75 (2019), pp. 1086–1097.

[52] J. LIN AND D.-X. ZHOU, Learning theory of randomized Kaczmarz algorithm, The Journal of Machine Learning Research, 16 (2015), pp. 3341–3365.

[53] C. LIU AND H. WECHSLER, Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition, IEEE Transactions on Image Processing, 11 (2002), pp. 467–476.

[54] J. LIU AND S. WRIGHT, An accelerated randomized Kaczmarz algorithm, Mathematics of Computation, 85 (2016), pp. 153–178.

[55] J. LIU, S. J. WRIGHT, AND S. SHIDHAR, An asynchronous parallel randomized Kaczmarz algorithm, arXiv preprint arXiv:1401.4780, (2014).

[56] P. MA, M. W. MAHONEY, AND B. YU, A statistical perspective on algorithmic leveraging, Journal of Machine Learning Research, 16 (2015), pp. 861–911.

[57] Q. MAI, A review of discriminant analysis in high dimensions, Wiley Interdisciplinary Reviews: Computational Statistics, 5 (2013), pp. 190–197.

[58] Q. MAI, H. ZOU, AND M. YUAN, A direct approach to sparse discriminant analysis in ultra-high dimensions, Biometrika, 99 (2012), pp. 29–42.

[59] O. MAILLARD AND R. MUNOS, Compressed least-squares regression, Advances in Neural Information Processing Systems, 22 (2009).

[60] T. P. MCDONALD, J. POERTNER, AND G. HARRIS, Predicting placement in foster care: A comparison of logistic regression and neural network analysis, Journal of social service research, 28 (2001), pp. 1–20.

[61] X. MENG, M. A. SAUNDERS, AND M. W. MAHONEY, LSRN: a parallel iterative solver for strongly over- or underdetermined systems, SIAM Journal on Scientific Computing, 36 (2014), pp. C95–C118.

[62] D. NEEDELL, Randomized Kaczmarz solver for noisy linear systems, BIT Numerical Mathematics, 50 (2010), pp. 395–403.

[63] D. NEEDELL, N. SREBRO, AND R. WARD, Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm, Mathematical Programming, 155 (2016), pp. 549–573.

[64] D. NEEDELL AND J. A. TROPP, Paved with good intentions: Analysis of a randomized block Kaczmarz method, Linear Algebra and its Applications, 441 (2014), pp. 199–221.

[65] D. NEEDELL AND R. WARD, Two-subspace projection method for coherent overdetermined systems, Journal of Fourier Analysis and Applications, 19 (2013), pp. 250–269.

[66] A. NEMIROVSKI, A. JUDITSKY, G. LAN, AND A. SHAPIRO, Robust stochastic approximation approach to stochastic programming, SIAM Journal on Optimization, 19 (2009), pp. 1574–
Contributors, *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, Nature Methods, 17 (2020), pp. 261–272, https://doi.org/10.1038/s41592-019-0686-2.

[91] 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 Artificial Intelligence and Statistics, PMLR, 2017, pp. 1150–1158.

[92] J. Wang, M. Tian, Y. Wang, G. He, and T. Liu, *Prkp: A parallel randomized iterative algorithm for solving linear systems*, in 2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), IEEE, 2022, pp. 244–249.

[93] T. Whitney and R. Meany, *Two algorithms related to the method of steepest descent*, SIAM journal on Numerical Analysis, 4 (1967), pp. 109–118.

[94] D. M. Witten and R. Tibshirani, *Penalized classification using Fisher’s linear discriminant*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 73 (2011), pp. 753–772.

[95] D. P. Woodruff et al., *Sketching as a tool for numerical linear algebra*, Foundations and Trends® in Theoretical Computer Science, 10 (2014), pp. 1–157.

[96] M. Xiong, L. Jin, W. Li, and E. Boerwinkle, *Computational methods for gene expression-based tumor classification*, Biotechniques, 29 (2000), pp. 1264–1270.

[97] Y. Yaniv, J. D. Moorman, W. Swartworth, T. Tu, D. Landis, and D. Needell, *Selectable set randomized Kaczmarz*, Numerical Linear Algebra with Applications, (2022). e2458.

[98] H. Ye, Y. Li, C. Chen, and Z. Zhang, *Fast fisher discriminant analysis with randomized algorithms*, Pattern Recognition, 72 (2017), pp. 82–92.

[99] J. Ye, *Least squares linear discriminant analysis*, in Proceedings of the 24th international conference on Machine learning, 2007, pp. 1087–1093.

[100] S. Zhou, L. Wasserman, and J. Lafferty, *Compressed regression*, Advances in Neural Information Processing Systems, 20 (2007).

[101] D. L. Zhu and P. Marcotte, *Co-coercivity and its role in the convergence of iterative schemes for solving variational inequalities*, SIAM Journal on Optimization, 6 (1996), pp. 714–726.

[102] L. Zhu, D. Qiu, D. Ergu, C. Ying, and K. Liu, *A study on predicting loan default based on the random forest algorithm*, Procedia Computer Science, 162 (2019), pp. 503–513.

[103] A. Zouzias and N. M. Freris, *Randomized extended Kaczmarz for solving least squares*, SIAM Journal on Matrix Analysis and Applications, 34 (2013), pp. 773–793.