Journal of Machine Learning Research 20 (2019) 1-41 Submitted 12/17; Revised 8/18; Published 2/19

Robust Frequent Directions with Application in Online Learning

Luo Luo RICKY@SJTU.EDU.CN
Cheng Chen JACK_CHEN1990@SJTU.EDU.CN
Department of Computer Science and Engineering
Shanghai Jiao Tong University
800 Dongchuan Road, Shanghai, China 200240

Zhihua Zhang* ZHZHANG@MATH.PKU.EDU.CN
National Engineering Lab for Big Data Analysis and Applications
School of Mathematical Sciences
Peking University
5 Yiheyuan Road, Beijing, China 100871

Wu-Jun Li LIWUJUN@NJU.EDU.CN
National Key Laboratory for Novel Software Technology
Collaborative Innovation Center of Novel Software Technology and Industrialization
Department of Computer Science and Technology
Nanjing University
163 Xianlin Avenue, Nanjing, China 210023

Tong Zhang TONGZHANG@TONGZHANG-ML.ORG
Computer Science & Mathematics
Hong Kong University of Science and Technology
Hong Kong

Editor: Qiang Liu

Abstract

The frequent directions (FD) technique is a deterministic approach for online sketching that has many applications in machine learning. The conventional FD is a heuristic procedure that often outputs rank deficient matrices. To overcome the rank deficiency problem, we propose a new sketching strategy called robust frequent directions (RFD) by introducing a regularization term. RFD can be derived from an optimization problem. It updates the sketch matrix and the regularization term adaptively and jointly. RFD reduces the approximation error of FD without increasing the computational cost. We also apply RFD to online learning and propose an effective hyperparameter-free online Newton algorithm. We derive a regret bound for our online Newton algorithm based on RFD, which guarantees the robustness of the algorithm. The experimental studies demonstrate that the proposed method outperforms state-of-the-art second order online learning algorithms.

Keywords: Matrix approximation, sketching, frequent directions, online convex optimization, online Newton algorithm

1. Introduction

The sketching technique is a powerful tool to deal with large scale matrices (Ghashami et al., 2016; Halko et al., 2011; Woodruff, 2014), and it has been widely used to speed up machine learning

---

*. Corresponding author.

algorithms such as second order optimization algorithms (Erdogdu and Montanari, 2015; Luo et al., 2016; Pilanci and Wainwright, 2017; Roosta-Khorasani and Mahoney, 2016a,b; Xu et al., 2016; Ye et al., 2017). There exist several families of matrix sketching strategies, including sparsification, column sampling, random projection (Achlioptas, 2003; Indyk and Motwani, 1998; Kane and Nelson, 2014; Wang et al., 2016), and frequent directions (FD) (Desai et al., 2016; Ghashami et al., 2016; Huang, 2018; Liberty, 2013; Mroueh et al., 2017; Ye et al., 2016).

Sparsification techniques (Achlioptas and McSherry, 2007; Achlioptas et al., 2013; Arora et al., 2006; Drineas and Zouzias, 2011) generate a sparse version of the matrix by element-wise sampling, which allows the matrix multiplication to be more efficient with lower space. Column (row) sampling algorithms (Mahoney, 2011) include the importance sampling (Drineas et al., 2006a,b; Frieze et al., 2004) and leverage score sampling (Drineas et al., 2012, 2008; Papailiopoulos et al., 2014). They define a probability for each row (column) and select a subset by the probability to construct the estimation. Random projection maps the rows (columns) of the matrix into lower dimensional space by a projection matrix. The projection matrix can be constructed in various ways (Woodruff, 2014) such as Gaussian random projections (Johnson and Lindenstrauss, 1984; Sarlos, 2006), fast Johnson-Lindenstrauss transforms (Ailon and Chazelle, 2006; Ailon and Liberty, 2009, 2013; Kane and Nelson, 2014) and sparse random projections (Clarkson and Woodruff, 2013; Nelson and Nguyên, 2013). The frequent directions (Ghashami et al., 2016; Liberty, 2013) is a deterministic sketching algorithm and achieves optimal tradeoff between approximation error and space.

In this paper we are especially concerned with the FD sketching (Desai et al., 2016; Ghashami et al., 2016; Liberty, 2013), because it is a stable online sketching approach. FD considers the matrix approximation in the streaming setting. In this case, the data is available in a sequential order and should be processed immediately. Typically, streaming algorithms can only use limited memory at any time. The FD algorithm extends the method of frequent items (Misra and Gries, 1982) to matrix approximation and has tight approximation error bound. However, FD usually leads to a rank deficient approximation, which in turn makes its applications less robust. For example, Newton-type algorithms require a non-singular and well-conditioned approximated Hessian matrix but FD sketching usually generates low-rank matrices. An intuitive and simple way to conquer this gap is to introduce a regularization term to enforce the matrix to be invertible (Luo et al., 2016; Roosta-Khorasani and Mahoney, 2016a,b). Typically, the regularization parameter is regarded as a hyperparameter and its choice is separable from the sketching procedure. Since the regularization parameter affects the the performance heavily in practice, it should be chosen carefully.

To overcome the weakness of the FD algorithm, we propose a new sketching approach that we call *robust frequent directions* (RFD). Unlike conventional sketching methods which only approximate the matrix with a low-rank structure, RFD constructs the low-rank part and updates the regularization term simultaneously. In particular, the update procedure of RFD can be regarded as solving an optimization problem (see Theorem 2). This method is different from the standard FD, giving rise to a tighter error bound.

Note that Zhang (2014) proposed matrix ridge approximation (MRA) to approximate a positive semi-definite matrix using an idea similar to RFD. There are two main differences between RFD and MRA. First, RFD is designed for the case that data samples come sequentially and memory is limited, while MRA has to access the whole data set. Second, MRA aims to minimize the approximation error with respect to the Frobenius norm while RFD tries to minimize the spectral-norm approximation error. In general, the spectral norm error bound is more meaningful than the Frobenius norm error bound (Tropp, 2015).

In a recent study, Luo et al. (2016) proposed a FD-based sketched online Newton (SON) algorithm (FD-SON) to accelerate the standard online Newton algorithms. Owing to the shortcoming of FD, the performance of FD-SON is significantly affected by the choice of the hyperparameter. Naturally, we can leverage RFD to improve online Newton algorithms. Accordingly, we propose a sketched online Newton step based on RFD (RFD-SON). Different from conventional sketched Newton algorithms, RFD-SON is hyperparameter-free. Setting the regularization parameter to be zero initially, RFD-SON will adaptively increase the regularization term. The approximation Hessian will be well-conditioned after a few iterations. Moreover, we prove that RFD-SON has a more robust regret bound than FD-SON, and the experimental results also validate better performance of RFD-SON.

The remainder of the paper is organized as follows. In Section 2 we present notation and preliminaries. In Section 3 we review the background of second order online learning and its sketched variants. In Sections 4 and 5 we propose our robust frequent directions (RFD) method and the applications in online learning, with some related theoretical analysis. In Section 6 we demonstrate empirical comparisons with baselines on serval real-world data sets to show the superiority of our algorithms. Finally, we conclude our work in Section 7.

2. Notation and Preliminaries

We let $\mathbf{I}_d$ denote the $d \times d$ identity matrix. For a matrix $\mathbf{A} = [A_{ij}] \in \mathbb{R}^{n \times d}$ of rank r where $r \leq \min(n, d)$, we let the condensed singular value decomposition (SVD) of $\mathbf{A}$ be $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ where $\mathbf{U} \in \mathbb{R}^{n \times r}$ and $\mathbf{V} \in \mathbb{R}^{d \times r}$ are column orthogonal and $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$ with $\sigma_1 \geq \sigma_2 \geq \ldots, \geq \sigma_r > 0$ places the nonzero singular values on its diagonal entries.

We use $\sigma_{\max}(\mathbf{A})$ to denote the largest singular value and $\sigma_{\min}(\mathbf{A})$ to denote the smallest non-zero singular value. Thus, the condition number of $\mathbf{A}$ is $\kappa(\mathbf{A}) = \frac{\sigma_{\max}(\mathbf{A})}{\sigma_{\min}(\mathbf{A})}$. The matrix pseudoinverse of $\mathbf{A}$ is defined by $\mathbf{A}^\dagger = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^\top \in \mathbb{R}^{d \times n}$.

Additionally, we let $\|\mathbf{A}\|_F = \sqrt{\sum_{i,j} A_{ij}^2} = \sqrt{\sum_{i=1}^r \sigma_i^2}$ be the Frobenius norm and $\|\mathbf{A}\|_2 = \sigma_{\max}(\mathbf{A})$ be the spectral norm. A matrix norm $|\!|\!| \cdot |\!|\!|$ is said to be unitarily invariant if $|\!|\!|\mathbf{PAQ}|\!|\!| = |\!|\!|\mathbf{A}|\!|\!|$ for any unitary matrices $\mathbf{P} \in \mathbb{R}^{n \times n}$ and $\mathbf{Q} \in \mathbb{R}^{d \times d}$. It is easy to verify that both the Frobenius norm and spectral norm are unitarily invariant. We define $[\mathbf{A}]_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^\top$ for $k \leq r$, where $\mathbf{U}_k \in \mathbb{R}^{n \times k}$ and $\mathbf{V}_k \in \mathbb{R}^{d \times k}$ are the first k columns of $\mathbf{U}$ and $\mathbf{V}$, and $\mathbf{\Sigma}_k = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k) \in \mathbb{R}^{k \times k}$. Then $[\mathbf{A}]_k$ is the best rank-k approximation to $\mathbf{A}$ in both the Frobenius and spectral norms, that is,

$$[\mathbf{A}]_k = \underset{\mathrm{rank}(\widehat{\mathbf{A}}) \leq k}{\mathrm{argmin}} \; \|\mathbf{A} - \widehat{\mathbf{A}}\|_F = \underset{\mathrm{rank}(\widehat{\mathbf{A}}) \leq k}{\mathrm{argmin}} \; \|\mathbf{A} - \widehat{\mathbf{A}}\|_2.$$

Given a positive semidefinite matrix $\mathbf{H} \in \mathbb{R}^{d \times d}$, the notation $\|\mathbf{x}\|_{\mathbf{H}}$ is called $\mathbf{H}$-norm of vector $\mathbf{x} \in \mathbb{R}^d$, that is, $\|\mathbf{x}\|_{\mathbf{H}} = \sqrt{\mathbf{x}^\top \mathbf{H} \mathbf{x}}$. If matrices $\mathbf{A}$ and $\mathbf{B}$ have the same size, we let $\langle \mathbf{A}, \mathbf{B} \rangle$ denote $\sum_{i,j} A_{ij} B_{ij}$.

2.1. Frequent Directions

We give a brief review of frequent directions (FD) (Ghashami et al., 2016; Liberty, 2013), because it is closely related to our proposed method. FD is a deterministic matrix sketching in the row-updates model. For any input matrix $\mathbf{A} \in \mathbb{R}^{T \times d}$ whose rows come sequentially, it maintains a sketch matrix $\mathbf{B} \in \mathbb{R}^{(m-1) \times d}$ with $m \ll T$ to approximate $\mathbf{A}^\top \mathbf{A}$ by $\mathbf{B}^\top \mathbf{B}$.

We present the detailed implementation of FD in Algorithm 1. The intuition behind FD is similar to that of frequent items. FD periodically shrinks orthogonal vectors by roughly the same amount (Line 5 of Algorithm 1). The shrinking step reduces the square Frobenius norm of the sketch reasonable and guarantees that no direction is reduced too much.

Algorithm 1 Frequent Directions

```
1: Input: $\mathbf{A} = [\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(T)}]^\top \in \mathbb{R}^{T \times d}$, $\mathbf{B}^{(m-1)} = [\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(m-1)}]^\top$
2: for $t = m, \dots, T$ do
3:     $\widehat{\mathbf{B}}^{(t-1)} = \begin{bmatrix} \mathbf{B}^{(t-1)} \\ (\mathbf{a}^{(t)})^\top \end{bmatrix}$
4:     Compute SVD: $\widehat{\mathbf{B}}^{(t-1)} = \mathbf{U}^{(t-1)} \mathbf{\Sigma}^{(t-1)} (\mathbf{V}^{(t-1)})^\top$
5:     $\mathbf{B}^{(t)} = \sqrt{\big(\mathbf{\Sigma}_{m-1}^{(t-1)}\big)^2 - \big(\sigma_m^{(t-1)}\big)^2 \mathbf{I}_{m-1}} \cdot \big(\mathbf{V}_{m-1}^{(t-1)}\big)^\top$
6: end for
7: Output: $\mathbf{B} = \mathbf{B}^{(T)}$
```

FD has the following error bound for any $k < m$,

$$\|\mathbf{A}^\top \mathbf{A} - \mathbf{B}^\top \mathbf{B}\|_2 \le \frac{1}{m-k} \|\mathbf{A} - [\mathbf{A}]_k\|_F^2. \tag{1}$$

The above result means that the space complexity of FD is optimal regardless of streaming issues because any algorithm satisfying $\|\mathbf{A}^\top \mathbf{A} - \mathbf{B}^\top \mathbf{B}\|_2 \le \|\mathbf{A} - [\mathbf{A}]_k\|_F^2/(m-k)$ requires $\mathcal{O}(md)$ space to represent matrix $\mathbf{B}$ (Ghashami et al., 2016). The dominated computation of the algorithm is computating the SVD of $\widehat{\mathbf{B}}^{(t-1)}$, which costs $\mathcal{O}(m^2 d)$ by the standard SVD implementation. However, the total cost can be reduced from $\mathcal{O}(Tm^2 d)$ to $\mathcal{O}(Tmd)$ by doubling the space (Algorithm 4 in Appendix A) or using the Gu-Eisenstat procedure (Gu and Eisenstat, 1993).

Desai et al. (2016) proposed some extensions of FD. More specifically, Parameterized FD (PFD) uses an extra hyperparameter to describe the proportion of singular values shrunk in each iteration. PFD improves the performance empirically, but has worse error bound than FD by a constant. Compensative FD (CFD) modifies the output of FD by increasing the singular values and keeps the same error guarantees as FD.

3. Online Newton Methods

For ease of demonstrating our work, we would like to introduce sketching techniques in online learning scenarios. First of all, we introduce the background of convex online learning including online Newton step algorithms. Then we discuss the connection between online learning and sketched second order methods, which motivates us to propose a more robust sketching algorithm.

3.1. Convex Online Learning

Online learning is performed in a sequence of consecutive rounds (Shalev-Shwartz, 2011). We consider the problem of convex online optimization as follows. For a sequence of examples $\{\mathbf{x}^{(t)} \in \mathbb{R}^d\}$, and convex smooth loss functions $\{f_t : \mathcal{K}_t \to \mathbb{R}\}$ where $f_t(\mathbf{w}) \triangleq \ell_t(\mathbf{w}^\top \mathbf{x}^{(t)})$ and $\mathcal{K}_t \subset \mathbb{R}^d$

are convex compact sets, the learner outputs a predictor $\mathbf{w}^{(t)}$ and suffers the loss $f_t(\mathbf{w}^{(t)})$ at the t-th round. The cumulative regret at round T is defined as:

$$R_T(\mathbf{w}^*) = \sum_{t=1}^{T} f_t(\mathbf{w}^{(t)}) - \sum_{t=1}^{T} f_t(\mathbf{w}^*),$$

where $\mathbf{w}^* = \operatorname{argmin}_{\mathbf{w}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{w})$ and $\mathcal{K} = \bigcap_{t=1}^{T} \mathcal{K}_t$.

We make the following assumptions on the loss functions.

Assumption 1 *The loss functions ℓ_t satisfy $|\ell_t'(z)| \leq L$ whenever $|z| \leq C$, where L and C are positive constants.*

Assumption 2 *There exists a $\mu_t \leq 0$ such that for all $\mathbf{u}, \mathbf{w} \in \mathcal{K}$, we have*

$$f_t(\mathbf{w}) \geq f_t(\mathbf{u}) + \nabla f_t(\mathbf{u})^\top (\mathbf{w} - \mathbf{u}) + \frac{\mu_t}{2}\big(\nabla f_t(\mathbf{u})^\top (\mathbf{w} - \mathbf{u})\big)^2.$$

Note that for a loss function f_t whose domain and gradient have bounded diameter, holding Assumption 2 only requires the exp-concave property, which is more general than strong convexity (Hazan, 2016). For example, the square loss function $f_t(\mathbf{w}) = (\mathbf{w}^\top \mathbf{x}_t - y_t)^2$ satisfies Assumption 2 with $\mu_t = \frac{1}{8C^2}$ if the function is subject to constraints $|\mathbf{w}^\top \mathbf{x}_t| \leq C$ and $y_t \leq C$ (Luo et al., 2016), but it is not strongly convex.

One typical online learning algorithm is online gradient descent (OGD) (Hazan et al., 2007; Zinkevich, 2003). At the $(t+1)$-th round, OGD exploits the following update rules:

$$\begin{aligned} \mathbf{u}^{(t+1)} &= \mathbf{w}^{(t)} - \beta_t \mathbf{g}^{(t)}, \\ \mathbf{w}^{(t+1)} &= \operatorname*{argmin}_{\mathbf{w}\in\mathcal{K}_{t+1}} \|\mathbf{w} - \mathbf{u}^{(t+1)}\|, \end{aligned}$$

where $\mathbf{g}^{(t)} = \nabla f_t(\mathbf{w}^{(t)})$ and β_t is the learning rate. The algorithm has linear computation cost and achieves $\mathcal{O}(\frac{L^2}{H}\log T)$ regret bound for the H-strongly convex loss.

In this paper, we are more interested in online Newton step algorithms (Hazan et al., 2007; Luo et al., 2016). The standard online Newton step keeps the curvature information in the matrix $\mathbf{H}^{(t)} \in \mathbb{R}^{d\times d}$ sequentially and iterates as follows:

$$\begin{aligned} \mathbf{u}^{(t+1)} &= \mathbf{w}^{(t)} - \beta_t (\mathbf{H}^{(t)})^{-1} \mathbf{g}^{(t)}, \\ \mathbf{w}^{(t+1)} &= \operatorname*{argmin}_{\mathbf{w}\in\mathcal{K}_{t+1}} \|\mathbf{w} - \mathbf{u}^{(t+1)}\|_{\mathbf{H}^{(t)}}. \end{aligned} \tag{2}$$

The matrix $\mathbf{H}^{(t)}$ is constructed by the outer product of historical gradients (Duchi et al., 2011; Luo et al., 2016), such as

$$\mathbf{H}^{(t)} = \sum_{i=1}^{t} \mathbf{g}^{(i)} (\mathbf{g}^{(i)})^\top + \alpha_0 \mathbf{I}_d, \tag{3}$$

$$\text{or} \quad \mathbf{H}^{(t)} = \sum_{i=1}^{t} (\mu_t + \eta_t) \mathbf{g}^{(i)} (\mathbf{g}^{(i)})^\top + \alpha_0 \mathbf{I}_d, \tag{4}$$

where $\alpha_0 \geq 0$ is a fixed regularization parameter, μ_t is the constant in Assumption 2, and η_t is typically chosen as $\mathcal{O}(1/t)$. The second order algorithms enjoy logarithmical regret bound without the strongly convex assumption but require quadratical space and computation cost. Some variants of online Newton algorithms have been applied to optimize neural networks (Martens and Grosse, 2015; Grosse and Martens, 2016; Ba et al., 2017), but they do not provide theoretical guarantee on nonconvex cases.

3.2. Efficient Algorithms by Sketching

To make the online Newton step scalable, it is natural to use sketching techniques (Woodruff, 2014). The matrix $\mathbf{H}^{(t)}$ in online learning has the form $\mathbf{H}^{(t)} = (\mathbf{A}^{(t)})^\top \mathbf{A}^{(t)} + \alpha_0 \mathbf{I}_d$, where $\mathbf{A}^{(t)} \in \mathbb{R}^{t \times d}$ is the corresponding term of (3) or (4) such as

$$\mathbf{A}^{(t)} = [\mathbf{g}^{(1)}, \dots, \mathbf{g}^{(t)}]^\top, \quad \text{or} \quad \mathbf{A}^{(t)} = [\sqrt{\mu_1 + \eta_1}\mathbf{g}^{(1)}, \dots, \sqrt{\mu_t + \eta_t}\mathbf{g}^{(t)}]^\top.$$

The sketching algorithm employs an approximation of $(\mathbf{A}^{(t)})^\top \mathbf{A}^{(t)}$ by $(\mathbf{B}^{(t)})^\top \mathbf{B}^{(t)}$, where the sketch matrix $\mathbf{B}^{(t)} \in \mathbb{R}^{m \times d}$ is much smaller than $\mathbf{A}^{(t)}$ and $m \ll d$. Then we can use $(\mathbf{B}^{(t)})^\top \mathbf{B}^{(t)} + \alpha_0 \mathbf{I}_d$ to replace $\mathbf{H}^{(t)}$ in update (2) (Luo et al., 2016). By the Woodbury identity formula, we can reduce the computation of the update from $\mathcal{O}(d^2)$ to $\mathcal{O}(m^2 d)$ or $\mathcal{O}(md)$. There are several choices of sketching techniques, such as random projection (Achlioptas, 2003; Indyk and Motwani, 1998; Kane and Nelson, 2014), frequent directions (Ghashami et al., 2016; Liberty, 2013) and Oja's algorithm (Oja, 1982; Oja and Karhunen, 1985). However, all above methods treat α_0 as a given hyperparameter which is independent of the sketch matrix $\mathbf{B}^{(t)}$. In practice, the performance of sketched online Newton methods is sensitive to the choice of the hyperparamter α_0.

4. Robust Frequent Directions

In many machine learning applications such as online learning (Hazan and Arora, 2006; Hazan et al., 2007; Hazan, 2016; Luo et al., 2016), Gaussian process regression (Rasmussen and Williams, 2006) and kernel ridge regression (Drineas and Mahoney, 2005), we usually require an additional regularization term to make the matrix invertible and well-conditioned, while conventional sketching methods only focus on the low-rank approximation. On the other hand, the update of frequent directions is not optimal in the view of minimizing the approximation error in each iteration. Both of them motivate us to propose robust frequent directions (RFD) that incorporates the update of sketch matrix and the regularization term into one framework.

4.1. The Algorithm

The RFD approximates $\mathbf{A}^\top \mathbf{A}$ by $\mathbf{B}^\top \mathbf{B} + \alpha \mathbf{I}_d$ with $\alpha > 0$. We demonstrate the detailed implementation of RFD in Algorithm 2.

The main difference between RFD and conventional sketching algorithms is the additional term $\alpha \mathbf{I}_d$. We can directly use Algorithm 2 to approximate $\mathbf{A}^\top \mathbf{A}$ with $\alpha^{(m-1)} = \alpha_0 > 0$ if the target matrix is $\mathbf{A}^\top \mathbf{A} + \alpha_0 \mathbf{I}_d$. Compared with the standard FD, RFD only needs to maintain one extra variable $\alpha^{(t)}$ by scalar operations in each iteration, hence the cost of RFD is almost the same as FD. Because the value of $\alpha^{(t)}$ is typically increasing from the $(m+1)$-th round in practice, the resulting $\mathbf{B}^\top \mathbf{B} + \alpha \mathbf{I}_d$ is positive definite even the initial $\alpha^{(0)}$ is zero. Also, we can further accelerate the algorithm by doubling the space.

Algorithm 2 Robust Frequent Directions

```
**Input:** $\mathbf{A} = [\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(T)}]^\top \in \mathbb{R}^{T \times d}$, $\mathbf{B}^{(m-1)} = [\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(m-1)}]^\top$, $\alpha^{(m-1)} = 0$
**for** $t = m, \dots, T$ **do**
    $\widehat{\mathbf{B}}^{(t-1)} = \begin{bmatrix} \mathbf{B}^{(t-1)} \\ (\mathbf{a}^{(t)})^\top \end{bmatrix}$
    Compute SVD: $\widehat{\mathbf{B}}^{(t-1)} = \mathbf{U}^{(t-1)} \mathbf{\Sigma}^{(t-1)} (\mathbf{V}^{(t-1)})^\top$
    $\mathbf{B}^{(t)} = \sqrt{\big(\mathbf{\Sigma}_{m-1}^{(t-1)}\big)^2 - \big(\sigma_m^{(t-1)}\big)^2 \mathbf{I}_{m-1}} \cdot \big(\mathbf{V}_{m-1}^{(t-1)}\big)^\top$
    $\alpha^{(t)} = \alpha^{(t-1)} + \big(\sigma_m^{(t-1)}\big)^2 / 2$
**end for**
**Output:** $\mathbf{B} = \mathbf{B}^{(T)}$ and $\alpha = \alpha^{(T)}$.
```

4.2. Theoretical Analysis

Before demonstrating the theoretical results of RFD, we review FD from the aspect of low-rank approximation which provides a motivation to the design of our algorithm. At the t-th round iteration of FD (Algorithm 1), we have the matrix $\mathbf{B}^{(t-1)}$ which is used to approximate $(\mathbf{A}^{(t-1)})^\top \mathbf{A}^{(t-1)}$ by $(\mathbf{B}^{(t-1)})^\top \mathbf{B}^{(t-1)}$ and we aim to construct a new approximation which includes the new data $\mathbf{a}^{(t)}$, that is,

$$(\mathbf{B}^{(t)})^\top \mathbf{B}^{(t)} \approx (\mathbf{B}^{(t-1)})^\top \mathbf{B}^{(t-1)} + \mathbf{a}^{(t)} (\mathbf{a}^{(t)})^\top = (\widehat{\mathbf{B}}^{(t-1)})^\top \widehat{\mathbf{B}}^{(t-1)}. \tag{5}$$

The straightforward way to find $\mathbf{B}^{(t)}$ is to minimize the approximation error of (5) based on the spectral norm with low-rank constraint:

$$\mathbf{B}'^{(t)} = \operatorname*{argmin}_{\text{rank}(\mathbf{C}) = m-1} \big\| (\widehat{\mathbf{B}}^{(t-1)})^\top \widehat{\mathbf{B}}^{(t-1)} - \mathbf{C}^\top \mathbf{C} \big\|_2. \tag{6}$$

By the SVD of $\widehat{\mathbf{B}}^{(t-1)}$, we have the solution $\mathbf{B}'^{(t)} = \mathbf{\Sigma}_{m-1}^{(t-1)} \big(\mathbf{V}_{m-1}^{(t-1)}\big)^\top$. In this view, the update of FD

$$\mathbf{B}^{(t)} = \sqrt{\big(\mathbf{\Sigma}_{m-1}^{(t-1)}\big)^2 - \big(\sigma_m^{(t-1)}\big)^2 \mathbf{I}_{m-1}} \cdot \big(\mathbf{V}_{m-1}^{(t-1)}\big)^\top \tag{7}$$

looks imperfect, because it is not an optimal low-rank approximation. However, the shrinkage operation in (7) is necessary. If we take a greedy strategy (Brand, 2002; Hall et al., 1998; Levey and Lindenbaum, 2000; Ross et al., 2008) which directly replaces $\mathbf{B}^{(t)}$ with $\mathbf{B}'^{(t)}$ in FD, it will perform worse in some specific cases[1] and also has no valid global error bound like (1).

Hence, the question is: can we devise a method which enjoys the optimality in each step and maintains global tighter error bound in the same time? Fortunately, RFD is just such an algorithm holding both the properties. We now explain the update rule of RFD formally, and provide the approximation error bound. We first give the following theorem which plays an important role in our analysis.

---

[1] We provide an example in Appendix F.

Theorem 1 *Given a positive semi-definite matrix* $\mathbf{M} \in \mathbb{R}^{d\times d}$ *and a positive integer* $k < d$, *let* $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top$ *be the SVD of* $\mathbf{M}$. *Let* $\mathbf{U}_k$ *denote the matrix of the first* k *columns of* $\mathbf{U}$ *and* σ_k *be the* k*-th singular value of* $\mathbf{M}$. *Then the pair* $(\widehat{\mathbf{C}}, \widehat{\delta})$, *defined as*

$$\widehat{\mathbf{C}} = \mathbf{U}_k(\mathbf{\Sigma}_k - \xi\mathbf{I}_k)^{1/2}\mathbf{Q} \quad and \quad \widehat{\delta} = (\sigma_{k+1} + \sigma_d)/2$$

where $\xi \in [\sigma_d, \sigma_{k+1}]$ *and* $\mathbf{Q}$ *is an arbitrary* $k \times k$ *orthonormal matrix, is the global minimizer of*

$$\min_{\mathbf{C}\in\mathbb{R}^{d\times k},\delta\in\mathbb{R}} \|\mathbf{M} - (\mathbf{C}\mathbf{C}^\top + \delta\mathbf{I}_d)\|_2. \tag{8}$$

Additionally, we have

$$\|\mathbf{M} - (\widehat{\mathbf{C}}\widehat{\mathbf{C}}^\top + \widehat{\delta}\mathbf{I}_d)\|_2 \leq \|\mathbf{M} - \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{U}_k^\top\|_2,$$

and the equality holds if and only if $\operatorname{rank}(\mathbf{M}) \leq k$.

Theorem 1 provides the optimal solution with the closed form for matrix approximation with a regularization term. In the case of $\operatorname{rank}(\mathbf{M}) > k$, the approximation $\widehat{\mathbf{C}}\widehat{\mathbf{C}}^\top + \widehat{\delta}\mathbf{I}_d$ is full rank and has strictly lower spectral norm error than the rank-k truncated SVD. Note that Zhang (2014) has established the Frobenius norm based result about the optimal analysis[2].

Recall that in the streaming case, our goal is to approximate the concentration of historical approximation and current data at the t-th round. The following theorem shows that the update of RFD is optimal with respect to the spectral norm for each step.

Theorem 2 *Based on the updates in Algorithm 2, we have*

$$(\mathbf{B}^{(t)}, \alpha^{(t)}) = \operatorname*{argmin}_{\mathbf{B}\in\mathbb{R}^{d\times(m-1)},\alpha\in\mathbb{R}} \left\|(\widehat{\mathbf{B}}^{(t-1)})^\top\widehat{\mathbf{B}}^{(t-1)} + \alpha^{(t-1)}\mathbf{I}_d - (\mathbf{B}^\top\mathbf{B} + \alpha\mathbf{I}_d)\right\|_2. \tag{9}$$

Theorem 2 explains RFD from an optimization viewpoint. It shows that each step of RFD is optimal for current information. Based on this theorem, the update of the standard FD corresponds $(\mathbf{B}, \alpha) = (\mathbf{B}^{(t)}, 0)$, which is not the optimal solution of (9). Intuitively, the regularization term of RFD compensates each direction for the over reduction from the shrinkage operation of FD. Theorem 2 also implies RFD is an online extension to the approximation of Theorem 1. We can prove Theorem 2 by using Theorem 1 with $\mathbf{M} = (\widehat{\mathbf{B}}^{(t-1)})^\top\widehat{\mathbf{B}}^{(t-1)} + \alpha^{(t-1)}\mathbf{I}_d$. We defer the details to Appendix C.

RFD also enjoys a tighter approximation error than FD as the following theorem shows.

Theorem 3 *For any* $k < m$ *and using the notation of Algorithm 2, we have*

$$\left\|\mathbf{A}^\top\mathbf{A} - (\mathbf{B}^\top\mathbf{B} + \alpha\mathbf{I}_d)\right\|_2 \leq \frac{1}{2(m-k)}\|\mathbf{A} - [\mathbf{A}]_k\|_F^2, \tag{10}$$

where $[\mathbf{A}]_k$ *is the best rank-k approximation to* $\mathbf{A}$ *in both the Frobenius and spectral norms.*

The right-hand side of inequality (10) is the half of the one in (1), which means RFD reduces the approximation error significantly with only one extra scalar.

The real applications usually consider the matrix with a regularization term. Hence we also consider approximating the matrix $\mathbf{M} = \mathbf{A}^\top\mathbf{A} + \alpha_0\mathbf{I}_d$ where $\alpha_0 > 0$ and the rows of $\mathbf{A}$ are available

---

[2] We also give a concise proof for the result of Zhang (2014) in Appendix B.

in sequentially order. Suppose that the standard FD approximates $\mathbf{A}^\top\mathbf{A}$ by $\mathbf{B}^\top\mathbf{B}$. Then it estimates $\mathbf{M}$ as $\mathbf{M}_{\text{FD}} = \mathbf{B}^\top\mathbf{B} + \alpha_0\mathbf{I}_d$. Meanwhile, RFD generates the approximation $\mathbf{M}_{\text{RFD}} = \mathbf{B}^\top\mathbf{B} + \alpha\mathbf{I}_d$ by setting $\alpha^{(m-1)} = \alpha_0$. Theorem 4 shows that the condition number of $\mathbf{M}_{\text{RFD}}$ is better than $\mathbf{M}_{\text{FD}}$ and $\mathbf{M}$. In general, the equality in Theorem 4 usually can not be achieved for $t > m$ unless $(\mathbf{a}^{(t)})^\top$ lies in the row space of $\mathbf{B}^{(t-1)}$ exactly or the first t rows of $\mathbf{A}$ have perfect low rank structure. Hence RFD is more likely to generate a well-conditioned approximation than others.

Theorem 4 *With the notation of Algorithms 1 and 2, let* $\mathbf{M} = \mathbf{A}^\top\mathbf{A} + \alpha_0\mathbf{I}_d$, $\mathbf{M}_{\text{FD}} = \mathbf{B}^\top\mathbf{B} + \alpha_0\mathbf{I}_d$, $\mathbf{M}_{\text{RFD}} = \mathbf{B}^\top\mathbf{B} + \alpha\mathbf{I}_d$ *and* $\alpha^{(m-1)} = \alpha_0$, *where* $\alpha_0 > 0$ *is a fixed scalar. Then we have* $\kappa(\mathbf{M}_{\text{RFD}}) \leq \kappa(\mathbf{M}_{\text{FD}})$ *and* $\kappa(\mathbf{M}_{\text{RFD}}) \leq \kappa(\mathbf{M})$.

5. The Online Newton Step by RFD

We now present the sketched online Newton step by robust frequent directions (RFD-SON). The procedure is shown in Algorithm 3, which is similar to sketched online Newton step (SON) algorithms (Luo et al., 2016) but uses the new sketching method RFD. The matrix $\mathbf{H}^{(t)}$ in Line 10 will not be constructed explicitly in practice, which is only used to the ease of analysis. The updates of $\mathbf{u}^{(t)}$ and $\mathbf{w}^{(t)}$ can be finished in $\mathcal{O}(md)$ time and space complexity by the Woodbury identity. We demonstrate the details in Appendix . When d is large, RFD-SON is much efficient than the standard online Newton step with the full Hessian that requires $\mathcal{O}(d^2)$ both in time and space.

Note that we do not require the hyperparameter α_0 to be strictly positive in RFD-SON. In practice, RFD-SON always archives good performance by setting $\alpha_0 = 0$, which leads to a hyperparameter-free algorithm, while the existing SON algorithm needs to select α_0 carefully. We consider the general case that $\alpha_0 \geq 0$ in this section for the ease of analysis.

Theorem 5 *Let* $\mu = \min_{t=1}^T\{\mu_t\}$ *and* $\mathcal{K} = \bigcap_{t=1}^T \mathcal{K}_t$. *Then under Assumptions 1 and 2 for any* $\mathbf{w} \in \mathcal{K}$, *Algorithm 3 has the following regret for* $\alpha_0 > 0$

$$R_T(\mathbf{w}) \leq \frac{\alpha_0}{2}\|\mathbf{w}\|^2 + 2(CL)^2\sum_{t=1}^T \eta_t + \frac{m}{2(\mu+\eta_T)}\ln\Big(\frac{\operatorname{tr}\big((\mathbf{B}^{(T)})^\top\mathbf{B}^{(T)}\big)}{m\alpha_0} + \frac{\alpha^{(T)}}{\alpha_0}\Big) + \Omega_{\text{RFD}} \tag{11}$$

where

$$\Omega_{\text{RFD}} = \frac{d-m}{2(\mu+\eta_T)}\ln\frac{\alpha^{(T)}}{\alpha_0} + \frac{m}{4(\mu+\eta_T)}\sum_{t=1}^T \frac{(\sigma_m^{(t)})^2}{\alpha^{(t)}} + C^2\sum_{t=1}^T(\sigma_m^{(t-1)})^2.$$

We present the regret bound of RFD-SON for positive α_0 in Theorem 5. The term Ω_{RFD} in (11) is the main gap between RFD-SON and the standard online Newton step without sketching. Ω_{RFD} is dominated by the last term which can be bounded as (1). If we exploit the standard FD to sketched online Newton step (Luo et al., 2016) (FD-SON), the regret bound is similar to (11) but the gap will be

$$\Omega_{\text{FD}} = \frac{m\Omega_k}{2(m-k)(\mu+\eta_T)\alpha_0},$$

Algorithm 3 RFD for Online Newton Step

```
Input: $\alpha^{(0)} = \alpha_0 \geq 0$, $m < d$, $\eta_t = \mathcal{O}(1/t)$, $\mathbf{w}^{(0)} = \mathbf{0}^d$ and $\mathbf{B}^{(0)}$ be empty.
for $t = 0, \dots, T-1$ do
    Receive example $\mathbf{x}^{(t)}$, and loss function $f_t(\mathbf{w})$
    Predict the output of $\mathbf{x}^{(t)}$ by $\mathbf{w}^{(t)}$ and suffer loss $f_t(\mathbf{w}^{(t)})$
    $\mathbf{g}^{(t)} = \nabla f_t(\mathbf{w}^{(t)})$
    $\widehat{\mathbf{B}}^{(t-1)} = \begin{bmatrix} \mathbf{B}^{(t-1)} \\ (\sqrt{\mu_t + \eta_t}\mathbf{g}^{(t)})^\top \end{bmatrix}$
    Compute SVD: $\widehat{\mathbf{B}}^{(t-1)} = \mathbf{U}^{(t-1)}\mathbf{\Sigma}^{(t-1)}(\mathbf{V}^{(t-1)})^\top$
    $\mathbf{B}^{(t)} = \sqrt{\big(\mathbf{\Sigma}_{m-1}^{(t-1)}\big)^2 - \big(\sigma_m^{(t-1)}\big)^2\mathbf{I}_{m-1}} \cdot (\mathbf{V}_{m-1}^{(t-1)})^\top$
    $\alpha^{(t)} = \alpha^{(t-1)} + \big(\sigma_m^{(t-1)}\big)^2/2$
    $\mathbf{H}^{(t)} = (\mathbf{B}^{(t)})^\top\mathbf{B}^{(t)} + \alpha^{(t)}\mathbf{I}_d$
    $\mathbf{u}^{(t+1)} = \mathbf{w}^{(t)} - (\mathbf{H}^{(t)})^\dagger\mathbf{g}^{(t)}$
    $\mathbf{w}^{(t+1)} = \operatorname{argmin}_{\mathbf{w}\in\mathcal{K}_{t+1}} \|\mathbf{w} - \mathbf{u}^{(t+1)}\|_{\mathbf{H}^{(t)}}$
end for
```

where Ω_k plays the similar role to term $\sum_{t=1}^{T}(\sigma_m^{(t-1)})^2$ in RFD-SON and the detailed definition can be found in Luo et al. (2016). This result is heavily dependent on the hyperparameter α_0. If we increase the value of α_0, the gap $\Omega_{\rm FD}$ can be reduced but the term $\frac{\alpha_0}{2}\|\mathbf{w}\|^2$ in the bound will increase, and vice versa. In other words, we need to trade off $\frac{\alpha_0}{2}\|\mathbf{w}\|^2$ and $\Omega_{\rm FD}$ by tuning α_0 carefully. For RFD-SON, Theorem 5 implies that we can set α_0 be sufficiently small to reduce $\frac{\alpha_0}{2}\|\mathbf{w}\|^2$ and it has limited effect on the term $\Omega_{\rm RFD}$. The reason is that the first term of $\Omega_{\rm RFD}$ contains $\frac{1}{\alpha_0}$ in the logarithmic function and the second term contains $\alpha^{(t)} = \alpha_0 + \frac{1}{2}\sum_{i=1}^{t-1}(\sigma_m^{(i)})^2$ in the denominator. For large t, $\alpha^{(t)}$ is mainly dependent on $\sum_{i=1}^{t-1}(\sigma_m^{(i)})^2$, rather than α_0. Hence the regret bound of RFD-SON is much less sensitive to the hyperparameter α_0 than FD-SON. We have $\sum_{t=1}^{T}(\sigma_m^{(t-1)})^2 \leq \|\mathbf{A} - [\mathbf{A}]_k\|_F^2/(m-k)$ for $k < m$ by using (17) with $\mathbf{A} = \sum_{t=1}^{T}(\mu_t + \eta_t)\mathbf{g}^{(t)}(\mathbf{g}^{(t)})^\top$.

Consider RFD-SON with $\alpha_0 = 0$. Typically, the parameter $\alpha^{(t)}$ is zero at very few first iterations and increase to be strictly positive later. Hence the learning algorithm can be divided into two phases based on whether $\alpha^{(t)}$ is zero. Suppose that T' satisfies

$$\alpha^{(t)} \begin{cases} = 0, & t < T' \\ > 0, & t \geq T'. \end{cases} \tag{12}$$

Then the regret can be written as

$$R_T(\mathbf{w}) = R_{1:T'}(\mathbf{w}) + R_{(T'+1):T}(\mathbf{w}),$$

where

$$R_{1:T'}(\mathbf{w}) = \sum_{t=1}^{T'} f_t(\mathbf{w}^{(t)}) - \sum_{t=1}^{T'} f_t(\mathbf{w})$$

and

$$R_{(T'+1):T}(\mathbf{w}) = \sum_{t=T'+1}^{T} f_t(\mathbf{w}^{(t)}) - \sum_{t=T'+1}^{T'} f_t(\mathbf{w}).$$

The regret from the first T' iterations can be bounded by Theorem 6 and the bound of $R_{(T'+1):T}(\mathbf{w})$ can be derived by the similar proof of Theorem 5.

Theorem 6 *By the condition of Theorem 5, $\mu' = \max_{t=1}^{T'}\{\mu_t\}$ and letting $\alpha_0 = 0$, σ^* be the smallest nonzero singular values of $\sum_{t=1}^{T'} \mathbf{g}^{(t)}(\mathbf{g}^{(t)})^\top$ and T' satisfy (12), we have that the first T' iterations of Algorithm 3 has the following regret*

$$R_{1:T'}(\mathbf{w}) \leq 2(CL)^2 \sum_{t=1}^{T'} \eta_t + \frac{m-1}{2(\eta_1+\mu')} + \frac{m(m-1)}{2(\eta_1+\mu')} \ln\Big(1 + \frac{2\sum_{t=1}^{T'} \|\mathbf{g}^{(t)}\|_2^2}{(1+r)r\sigma^*}\Big). \tag{13}$$

Combining above results, we can conclude the regret bound for the hyperparameter-free algorithm in Theorem 7. In practice, we often set m to be much smaller than d and T, which implies T' is much smaller than T. Hence, the regret bound of Theorem 7 is similar to the one of Theorem 5 when α_0 is close to 0. We can use RFD-SON with $\alpha_0 = 0$ and $\eta_t = 1/t$ to obtain a hyperparameter-free sketched online Newton algorithm. Luo et al. (2016) have proposed a hyperparameter-free online Newton algorithm without sketching and their regret contains a term with coefficient d.

Theorem 7 *Consider Algorithm 3 with $\alpha_0 = 0$, let T' satisfy (12), $\mu = \min_{t=1}^{T}\{\mu_t\}$, $\mu' = \max_{t=1}^{T'}\{\mu_t\}$, $\mathcal{K} = \bigcap_{t=1}^{T}\mathcal{K}_t$, σ^* has the same definition of Theorem 6 and $\alpha_0' = \det(\mathbf{H}^{(T')})$. Then under Assumptions 1 and 2 for any $\mathbf{w} \in \mathcal{K}$, we have that*

$$\begin{aligned} R_T(\mathbf{w}) \leq\ & 2(CL)^2 \sum_{t=1}^{T} \eta_t + \frac{m-1}{2(\eta_1+\mu')} + \frac{m(m-1)}{2(\eta_1+\mu')} \ln\Big(1 + \frac{2\sum_{t=1}^{T'} \|\mathbf{g}^{(t)}\|_2^2}{(1+r)r\sigma^*}\Big) \\ & + \frac{1}{2}\|\mathbf{w}^{(T')}\|^2_{\mathbf{H}^{(T')}} + \frac{m}{2(\mu+\eta_T)} \ln\Big(\mathrm{tr}\frac{\big((\mathbf{B}^{(T)})^\top \mathbf{B}^{(T)}\big)}{m\alpha^{(T')}} + \frac{\alpha^{(T)}}{\alpha_0'}\Big) + \Omega'_{\mathrm{RFD}} \end{aligned} \tag{14}$$

where

$$\Omega'_{\mathrm{RFD}} = \frac{d-m}{2(\mu+\eta_T)} \ln\frac{\alpha^{(T)}}{\alpha_0'} + \frac{m}{4(\mu+\eta_T)} \sum_{t=T'+1}^{T} \frac{(\sigma_m^{(t)})^2}{\alpha^{(t)}} + C^2 \sum_{t=T'+1}^{T} (\sigma_m^{(t-1)})^2.$$

6. Experiments

In this section, we evaluate the performance of robust frequent directions (RFD) and online Newton step by RFD (RFD-SON) on six real-world data sets "a9a," "gisette," "sido0," "farm-ads," "rcv1" and "real-sim," whose details are listed in Table 1. The data sets "sido0" and "farm-ads"can be found on Causality Workbench[3], and UCI Machine Learning Repository[4]. The others can be downloaded from LIBSVM repository[5]. The experiments are conducted in Matlab and run on a server with Intel (R) Core (TM) i7-3770 CPU 3.40GHz×2, 8GB RAM and 64-bit Windows Server 2012 system.

data sets	n	d	source
a9a	32,561	123	(Platt, 1999)
gisette	6,000	5,000	(Guyon et al., 2004)
sido0	12,678	4,932	(Guyon et al., 2008)
farm-ads	4,143	54,877	(Mesterharm and Pazzani, 2011)
rcv1	20,242	47,236	(Lewis et al., 2004)
real-sim	72,309	20,958	(McCallum)

Table 1: Summary of data sets used in our experiments

6.1. Matrix Approximation

We evaluate the approximation errors of the deterministic sketching algorithms including frequent directions (FD) (Liberty, 2013; Ghashami et al., 2016), parameterized frequent directions (PFD), compensative frequent directions (CFD) (Desai et al., 2016) and robust frequent directions (RFD). For a given data set $\mathbf{A} \in \mathbb{R}^{n\times d}$ of n samples with d features, we use the accelerated algorithms (see details in Appendix A) to approximate the covariance matrix $\mathbf{A}^\top\mathbf{A}$ by $\mathbf{B}^\top\mathbf{B}$ for FD, PFD, CFD and by $\mathbf{B}^\top\mathbf{B} + \alpha\mathbf{I}_d$ for RFD, respectively. We measure the performance according to the relative spectral norm error. We report the relative spectral norm error by varying the sketch size m.

Figure 1 shows the performance of FD, CFD and RFD. These three methods have no extra hyperparameter and their outputs only rely on the sketch size. The relative error of RFD is always smaller than that of FD and CFD. The error of RFD is nearly half of the error of FD in most cases, which matches our theoretical analysis in Theorem 3 very well.

Figure 2 compares the performance of RFD and PFD with different choices of the hyperparameter. We use PFD-β to refer the PFD algorithm where $\lfloor \beta m \rfloor$ singular values will get affected by the shrinkage steps. The extra hyperparameter β is tuned from $\{0.2, 0.4, 0.6, 0.8\}$. The result shows that RFD is better than PFD in most cases. PFD sometimes can achieve lower approximation error with a good choice of β. However, selecting the hyperparameter requires additional computation.

6.2. Online Learning

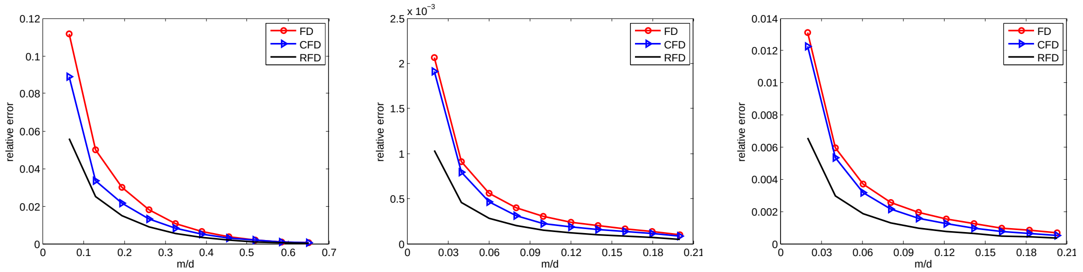
We now evaluate the performance of RFD-SON. We use the least squares loss $f_t(\mathbf{w}) = (\mathbf{w}^\top\mathbf{x}^{(t)} - y^{(t)})^2$, and set $\mathcal{K}_t = \{\mathbf{w} : |\mathbf{w}^\top\mathbf{x}^{(t)}| \leq 1\}$. In the experiments, we use the doubling space strategy (Algorithm 5 in Appendix A). We use 70% of the data set for training and the rest for test. The algorithms in the experiments include ADAGRAD, the standard online Newton step with the full Hessian (Duchi et al., 2011) (FULL-ON), the sketched online Newton step with frequent directions (FD-SON), the parameterized frequent directions (PFD-SON), the random projections (RP-SON), Oja's algorithms (Oja-SON) (Luo et al., 2016; Desai et al., 2016), and our proposed sketched online Newton step with RFD (RFD-SON).

---

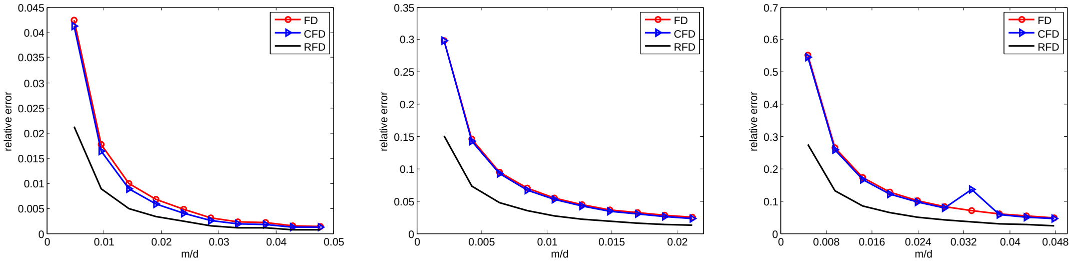
[3] https://www.causality.inf.ethz.ch/data/SIDO.html

[4] https://archive.ics.uci.edu/ml/datasets/Farm+Ads

[5] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/

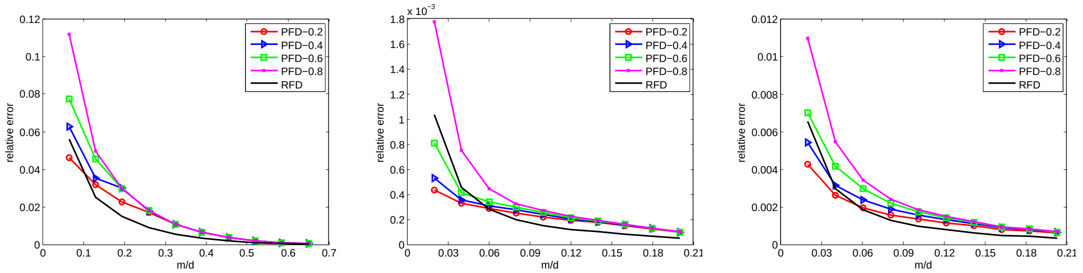


(a) a9a (b) gisette (c) sido0

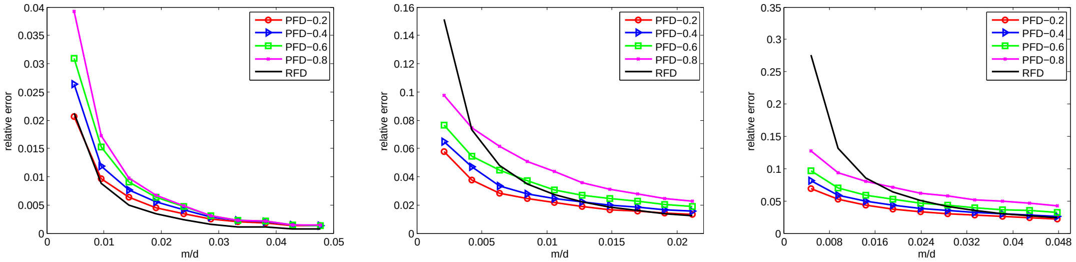


(d) farm-ads (e) rcv1 (f) real-sim

Figure 1: Comparison of relative spectral error of FD, CFD and RFD with proportion of sketching



(a) a9a (b) gisette (c) sido0



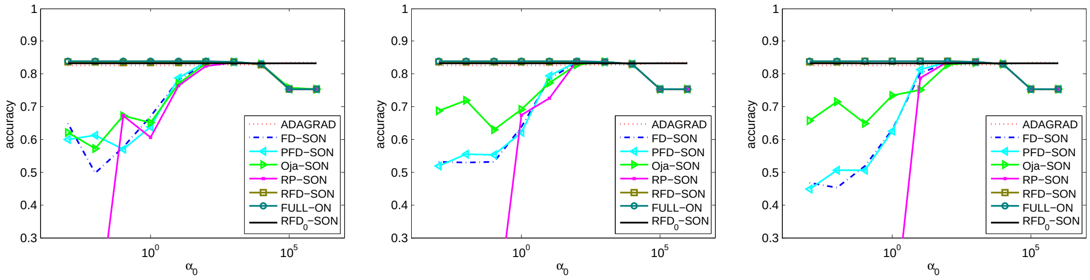
(d) farm-ads (e) rcv1 (f) real-sim

Figure 2: Comparison of relative spectral error of PFD and RFD with proportion of sketching

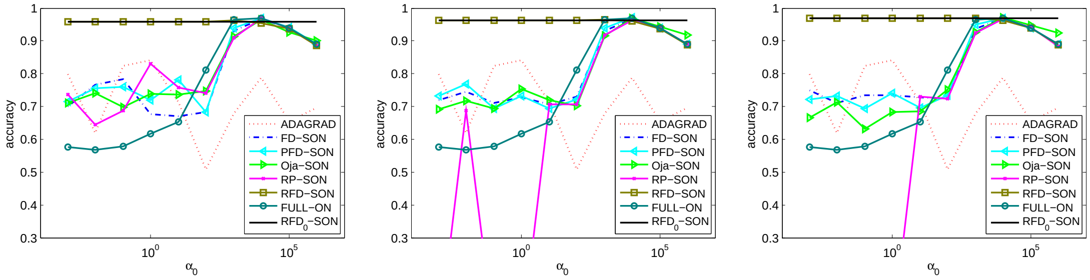
The hyperparameter α_0 is tuned from $\{10^{-3}, 10^{-2} \dots\ 10^5, 10^6\}$ for all methods and let $\eta = 1/t$ for SON algorithms. FULL-ON is too expensive and impractical for large d, so we exclude it from experiments on "farm-ads," "rcv1" and "real-sim." For PFD-SON, we let $\beta = 0.2$ heuristically because it usually achieves good performance on approximating the covariance matrix. Additionally, RFD-SON includes the result with $\alpha_0 = 0$ (RFD$_0$-SON). The sketch size of sketched online Newton methods is chosen from $\{5, 10, 20\}$ for "a9a," "gisette," "sido0," and $\{20, 30, 50\}$ for "farm-ads," "rcv1" and "real-sim." We measure performance according to two metrics (Duchi et al., 2011): the online error rate and the test set performance of the predictor at the end of one pass through the training data.

We are interested in how the hyperparameter α_0 affects the performance of the algorithms. We display the test set performance in Figures 3 and 4. We compare the online error rate of RFD$_0$-SON with the one of FULL-ON in Figure 5 and show the comparison between RFD$_0$-SON and other SON methods with different choices of α_0 in Figures 6 - 11.

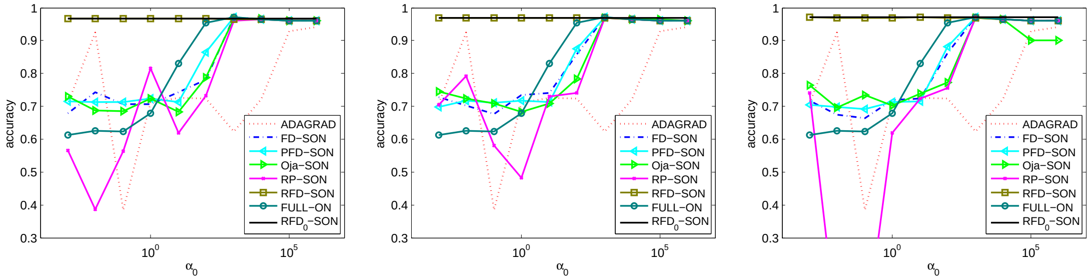
We also report the accuracy on the test sets for all algorithms at one pass with the best α_0 in Table 2 and the corresponding running times in Table 3. All SON algorithms can perform well with the best choice of α_0. However, only RFD$_0$-SON can perform well without tuning the hyperparameter while all baseline methods ADAGRAD, FD-SON, PFD-SON, RP-SON and Oja-SON are very sensitive to the value of α_0. The sub-figure (j)-(l) in Figures 5-11 shows RFD-SON usually has good performance with small α_0, which validates our theoretical analysis in Theorem 5. The choice of the hyperparameter almost has no effect of RFD-SON on data set “a9a’,’ “gisette,” “sido0” and “farm-ads.” These results verify that RFD-SON is a very stable algorithm in practice.



(a) a9a, $m = 5$ (b) a9a, $m = 10$ (c) a9a, $m = 20$

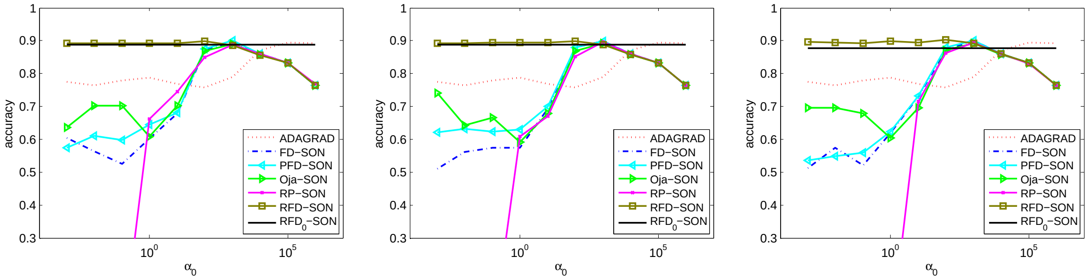


(d) gisette, $m = 5$ (e) gisette, $m = 10$ (f) gisette, $m = 20$

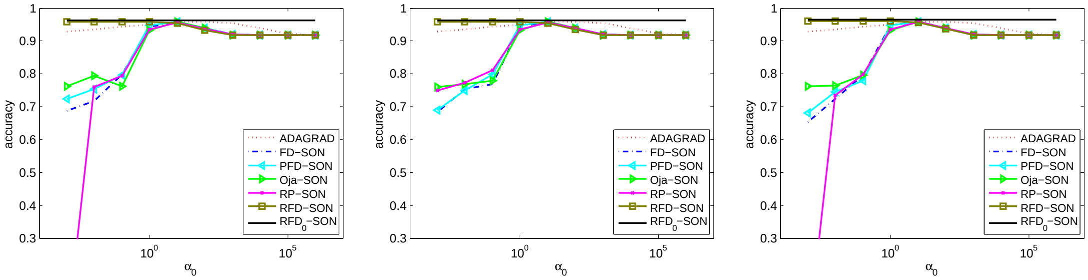


(g) sido0, $m = 5$ (h) sido0, $m = 10$ (i) sido0, $m = 20$

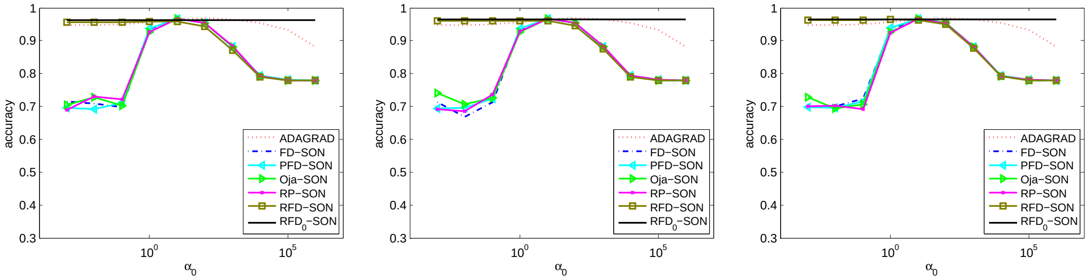
Figure 3: Comparison of the test error at the end of one pass on “a9a”, “gisette”, “sido0”



(a) farm-ads, $m = 20$ (b) farm-ads, $m = 30$ (c) farm-ads, $m = 50$

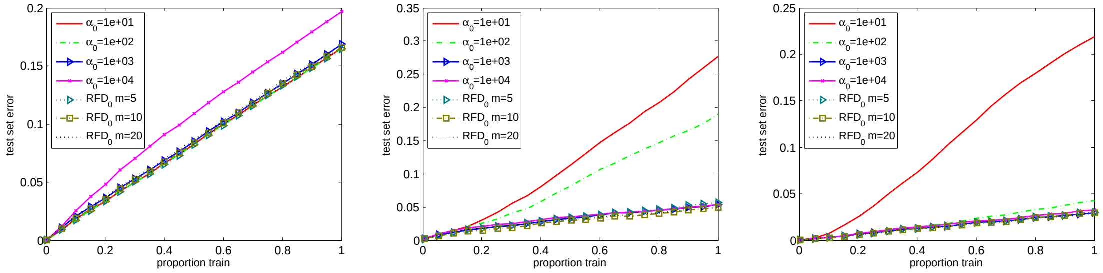


(d) rcv1, $m = 20$ (e) rcv1, $m = 30$ (f) rcv1, $m = 50$



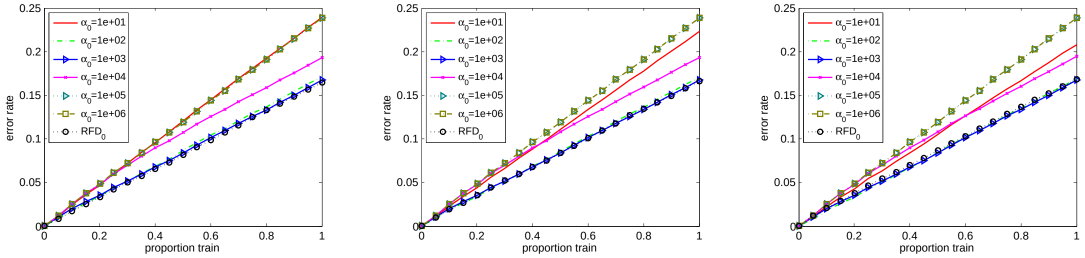
(g) real-sim, $m = 20$ (h) real-sim, $m = 30$ (i) real-sim, $m = 50$

Figure 4: Comparison of the test error at the end of one pass on “farm-ads”, “rcv1”, “real-sim”

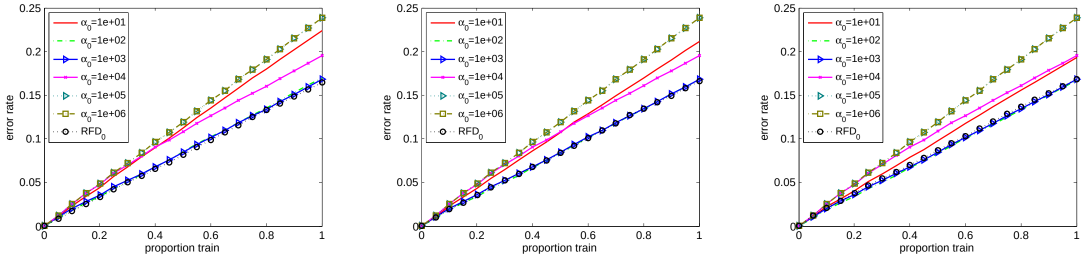


(a) a9a (a) gisette (c) sido0

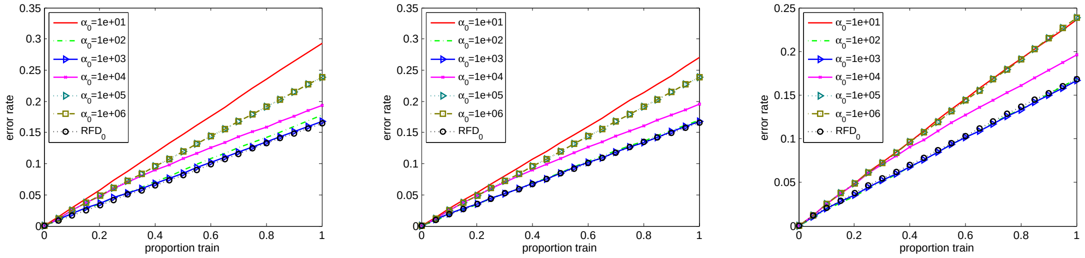
Figure 5: Comparison of the online error rate between algorithm FULL-ON and RFD$_0$



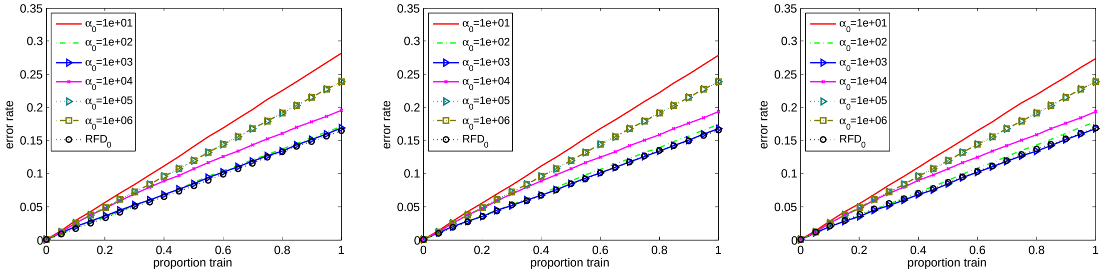
(a) FD vs RFD$_0$, $m = 5$ (b) FD vs RFD$_0$, $m = 10$ (c) FD vs RFD$_0$, $m = 20$



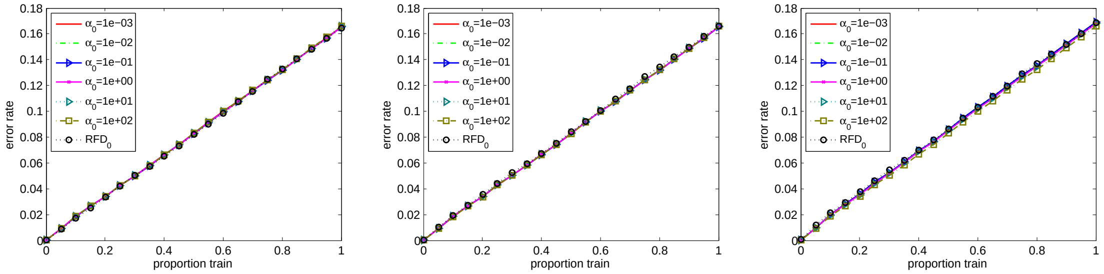
(a) PFD vs RFD$_0$, $m = 5$ (b) PFD vs RFD$_0$, $m = 10$ (c) PFD vs RFD$_0$, $m = 20$



(d) RP vs RFD$_0$, $m = 5$ (e) RP vs RFD$_0$, $m = 10$ (f) RP vs RFD$_0$, $m = 20$

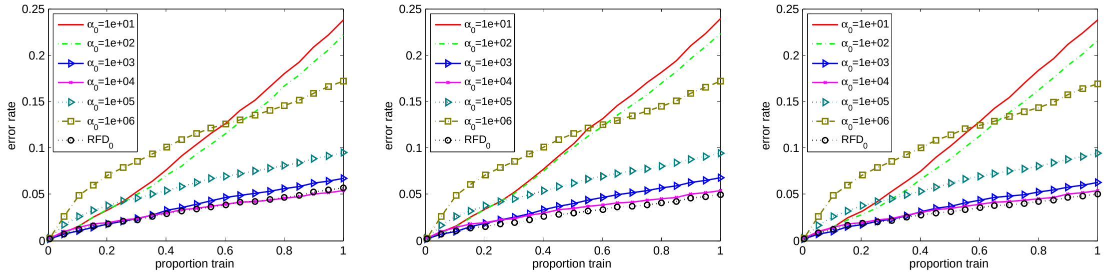


(g) Oja vs RFD$_0$, $m = 5$ (h) Oja vs RFD$_0$, $m = 10$ (i) Oja vs RFD$_0$, $m = 20$

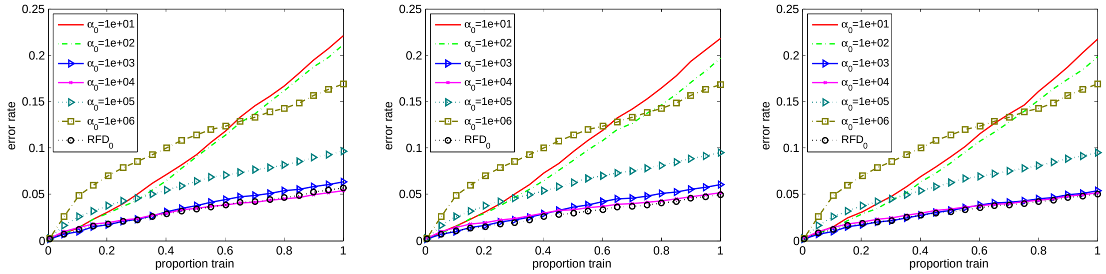


(j) RFD vs RFD$_0$, $m = 5$ (k) RFD vs RFD$_0$, $m = 10$ (l) RFD vs RFD$_0$, $m = 20$

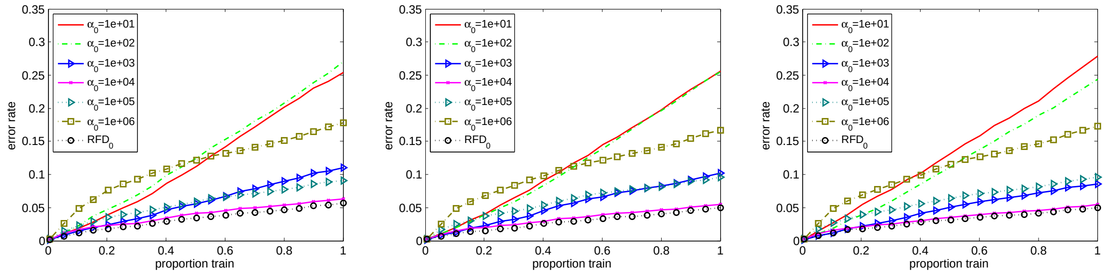
Figure 6: Comparison of the online error rate on “a9a”



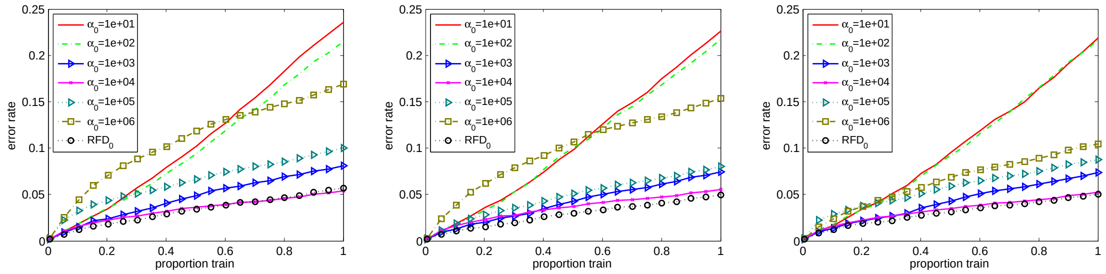
(a) FD vs RFD$_0$, $m = 5$ (b) FD vs RFD$_0$, $m = 10$ (c) FD vs RFD$_0$, $m = 20$



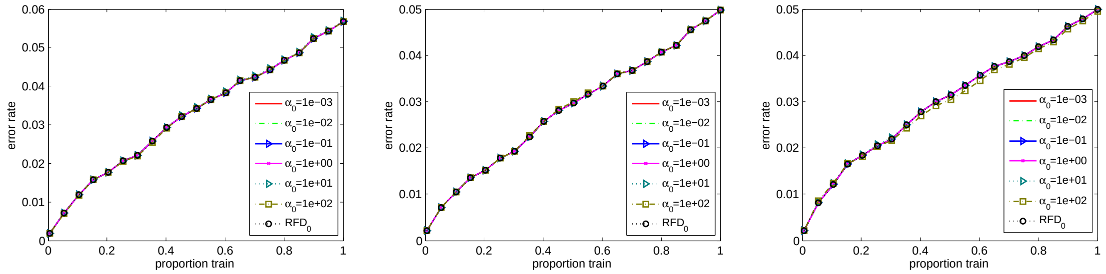
(a) PFD vs RFD$_0$, $m = 5$ (b) PFD vs RFD$_0$, $m = 10$ (c) PFD vs RFD$_0$, $m = 20$



(d) RP vs RFD$_0$, $m = 5$ (e) RP vs RFD$_0$, $m = 10$ (f) RP vs RFD$_0$, $m = 20$

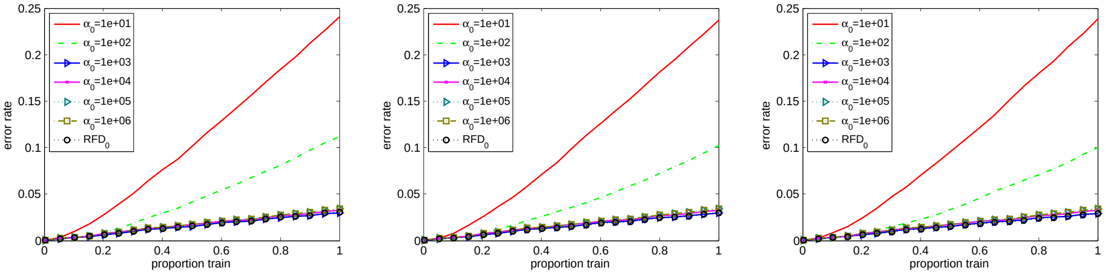


(g) Oja vs RFD$_0$, $m = 5$ (h) Oja vs RFD$_0$, $m = 10$ (i) Oja vs RFD$_0$, $m = 20$

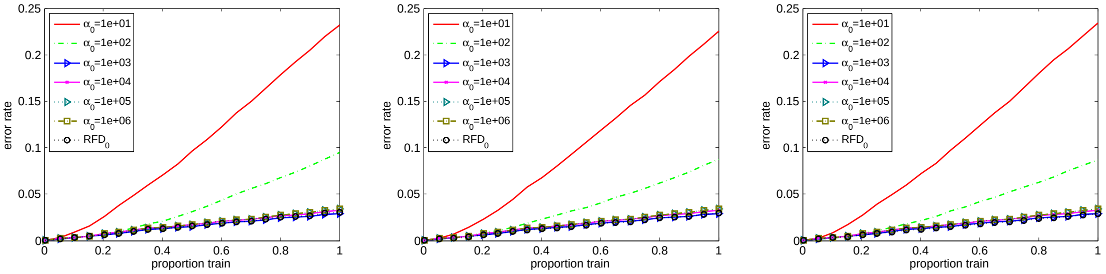


(j) RFD vs RFD$_0$, $m = 5$ (k) RFD vs RFD$_0$, $m = 10$ (l) RFD vs RFD$_0$, $m = 20$

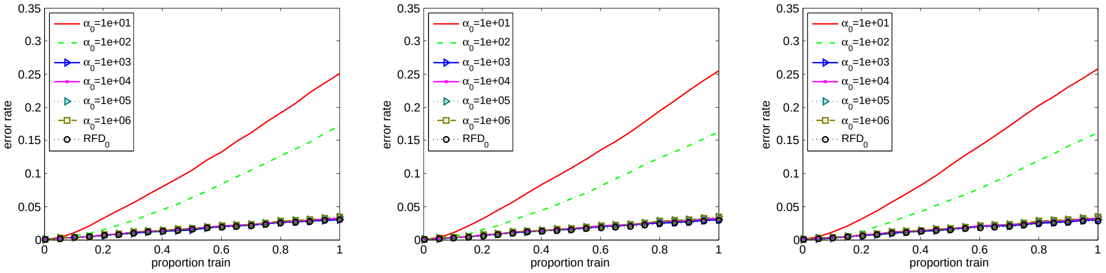
Figure 7: Comparison of the online error rate on “gisette”



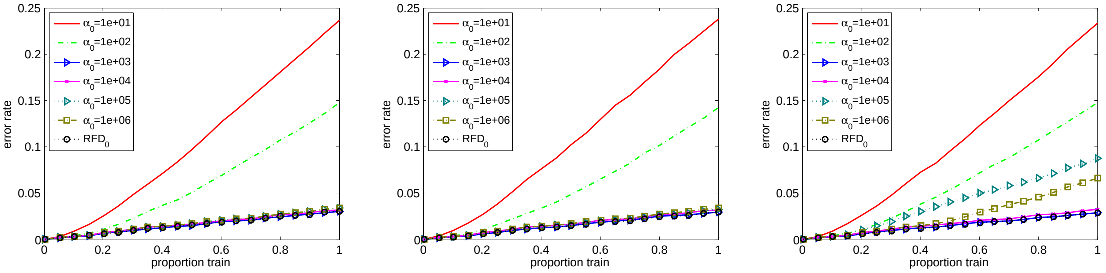
(a) FD vs RFD$_0$, $m = 5$ (b) FD vs RFD$_0$, $m = 10$ (c) FD vs RFD$_0$, $m = 20$



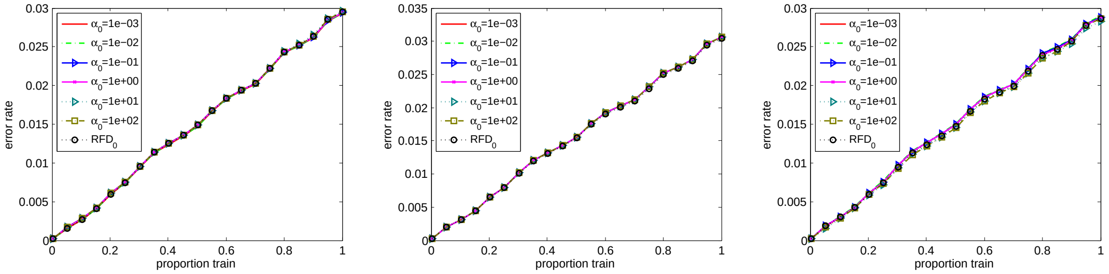
(a) PFD vs RFD$_0$, $m = 5$ (b) PFD vs RFD$_0$, $m = 10$ (c) PFD vs RFD$_0$, $m = 20$



(d) RP vs RFD$_0$, $m = 5$ (e) RP vs RFD$_0$, $m = 10$ (f) RP vs RFD$_0$, $m = 20$

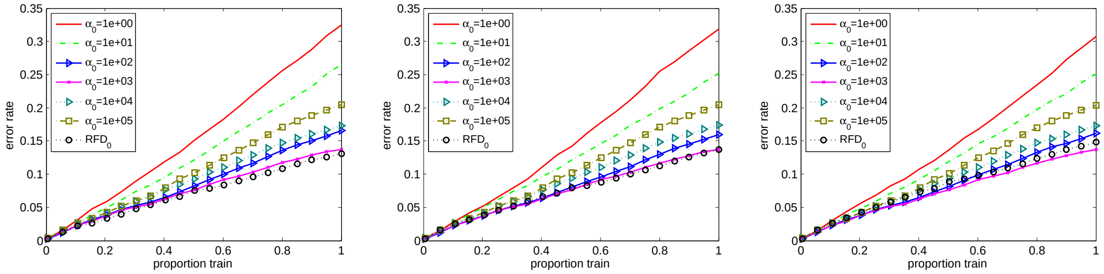


(g) Oja vs RFD$_0$, $m = 5$ (h) Oja vs RFD$_0$, $m = 10$ (i) Oja vs RFD$_0$, $m = 20$

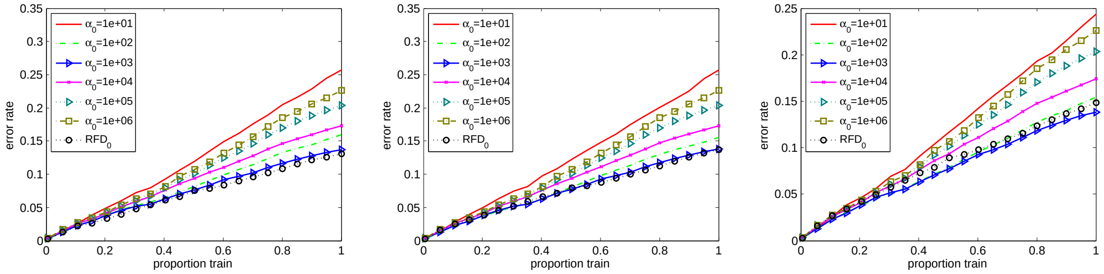


(j) RFD vs RFD$_0$, $m = 5$ (k) RFD vs RFD$_0$, $m = 10$ (l) RFD vs RFD$_0$, $m = 20$

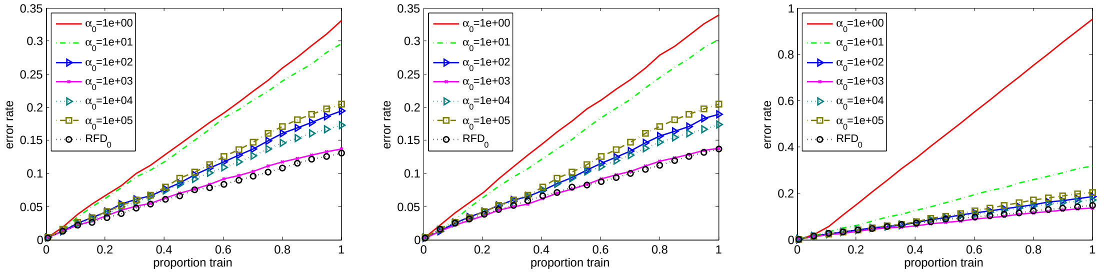
Figure 8: Comparison of the online error rate on “sido0”



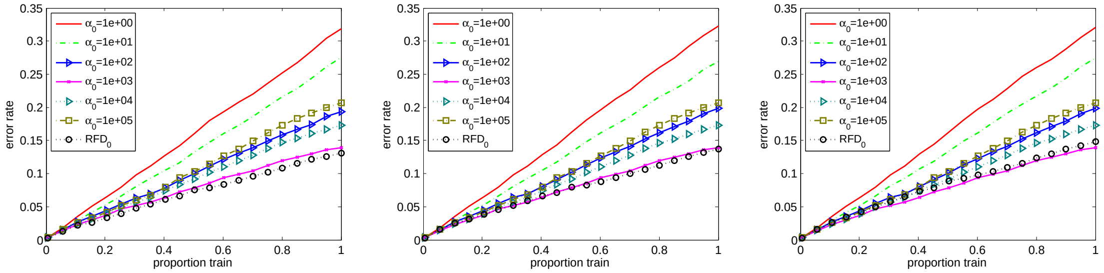
(a) FD vs RFD$_0$, $m = 20$ (b) FD vs RFD$_0$, $m = 30$ (c) FD vs RFD$_0$, $m = 50$



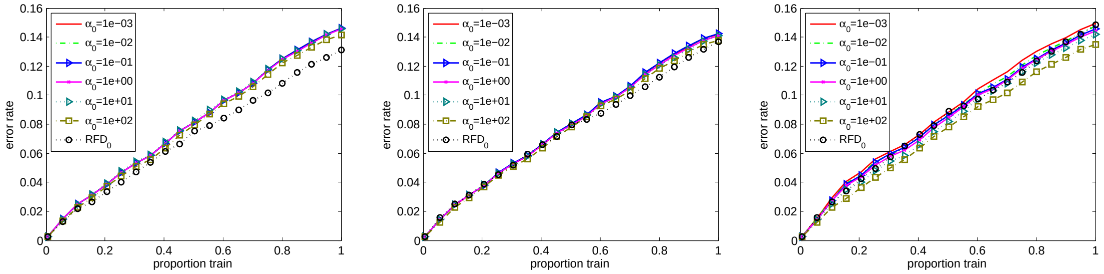
(a) PFD vs RFD$_0$, $m = 20$ (b) PFD vs RFD$_0$, $m = 30$ (c) PFD vs RFD$_0$, $m = 50$



(d) RP vs RFD$_0$, $m = 20$ (e) RP vs RFD$_0$, $m = 30$ (f) RP vs RFD$_0$, $m = 50$

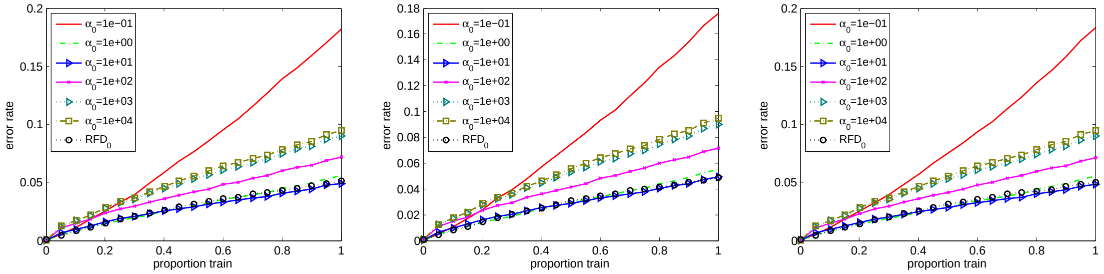


(g) Oja vs RFD$_0$, $m = 20$ (h) Oja vs RFD$_0$, $m = 30$ (i) Oja vs RFD$_0$, $m = 50$

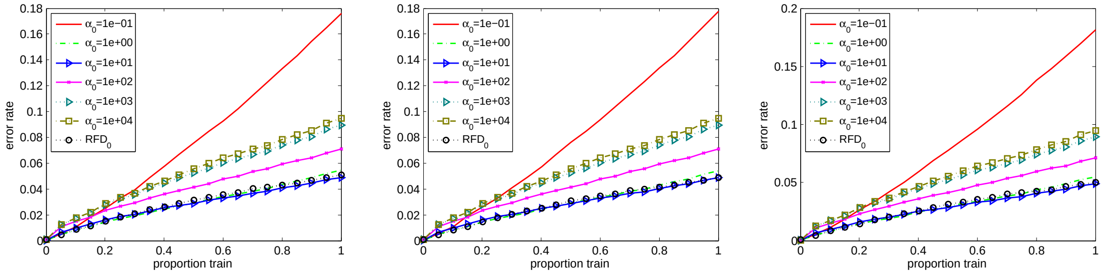


(j) RFD vs RFD$_0$, $m = 20$ (k) RFD vs RFD$_0$, $m = 30$ (l) RFD vs RFD$_0$, $m = 50$

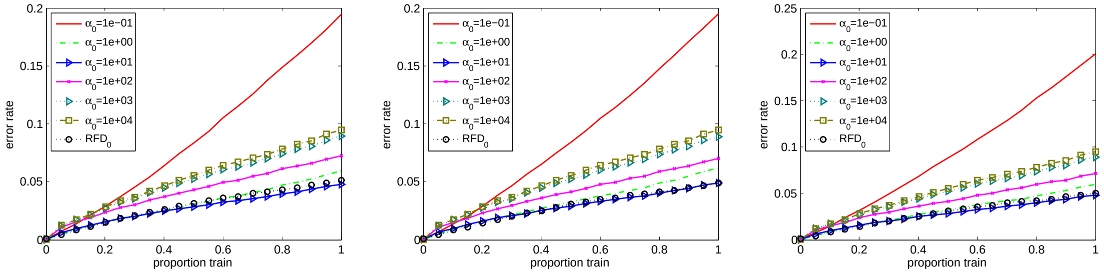
Figure 9: Comparison of the online error rate on “farm-ads”



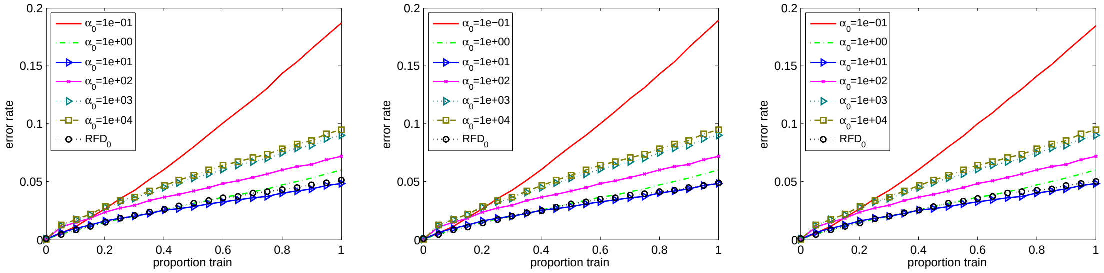
(a) FD vs RFD_0, $m = 20$ (b) FD vs RFD_0, $m = 30$ (c) FD vs RFD_0, $m = 50$



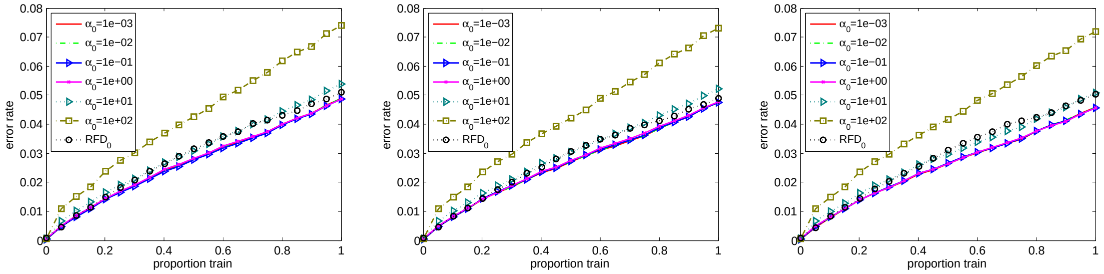
(a) PFD vs RFD_0, $m = 20$ (b) FD vs PFD_0, $m = 30$ (c) PFD vs RFD_0, $m = 50$



(d) RP vs RFD_0, $m = 20$ (e) RP vs RFD_0, $m = 30$ (f) RP vs RFD_0, $m = 50$

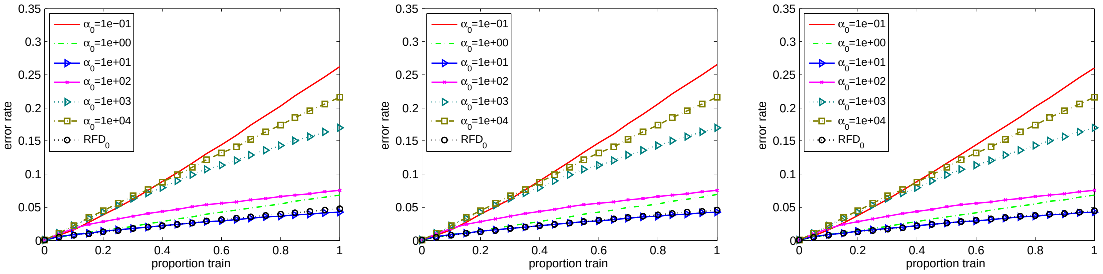


(g) Oja vs RFD_0, $m = 20$ (h) Oja vs RFD_0, $m = 30$ (i) Oja vs RFD_0, $m = 50$

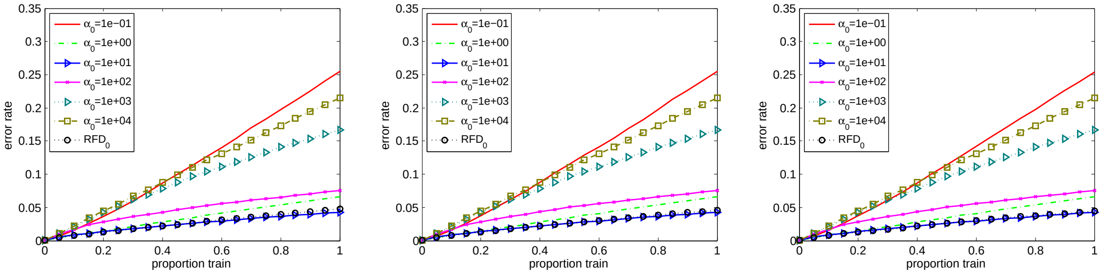


(j) RFD vs RFD_0, $m = 20$ (k) RFD vs RFD_0, $m = 30$ (l) RFD vs RFD_0, $m = 50$

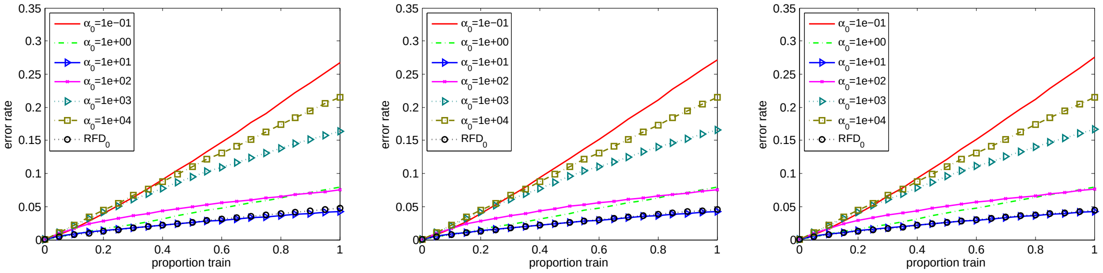
Figure 10: Comparison of the online error rate on “rcv1”



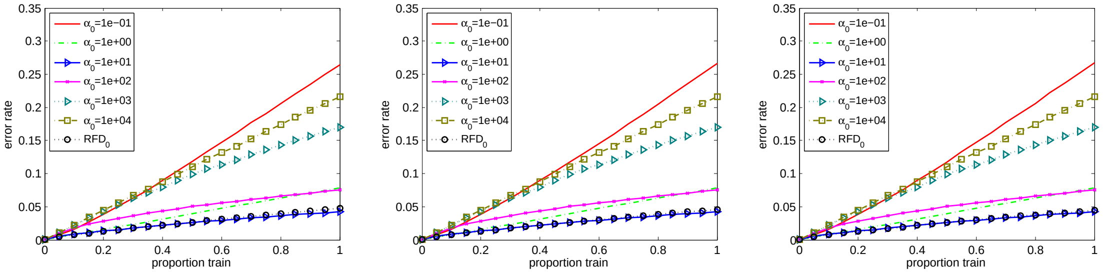
(a) FD vs RFD$_0$, $m = 20$ (b) FD vs RFD$_0$, $m = 30$ (c) FD vs RFD$_0$, $m = 50$



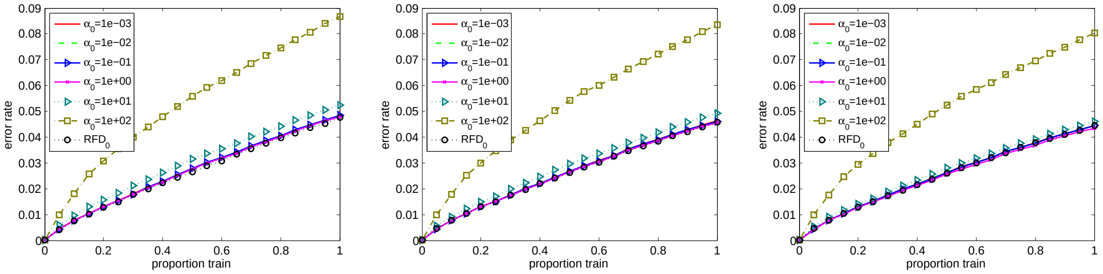
(a) PFD vs RFD$_0$, $m = 20$ (b) PFD vs RFD$_0$, $m = 30$ (c) PFD vs RFD$_0$, $m = 50$



(d) RP vs RFD$_0$, $m = 20$ (e) RP vs RFD$_0$, $m = 30$ (f) RP vs RFD$_0$, $m = 50$



(g) Oja vs RFD$_0$, $m = 20$ (h) Oja vs RFD$_0$, $m = 30$ (i) Oja vs RFD$_0$, $m = 50$



(j) RFD vs RFD$_0$, $m = 20$ (k) RFD vs RFD$_0$, $m = 30$ (l) RFD vs RFD$_0$, $m = 50$

Figure 11: Comparison of the online error rate on “real-sim”

Algorithms	a9a	gisette	sido0	farm-ads	rcv1	real-sim
ADAGRAD	83.4783	84.1111	94.0326	89.3805	95.8340	96.7315
FULL-ON	83.8264	96.9444	97.0557	/	/	/
FD, $m = m_1$	83.6524	96.7222	97.0557	89.7023	95.6200	96.7315
FD, $m = m_2$	83.6728	96.7222	97.0557	89.9437	95.6529	96.7361
FD, $m = m_3$	83.6728	96.7222	97.0557	89.9437	95.6529	96.7361
PFD, $m = m_1$	83.6728	97.0000	97.0294	90.0241	95.8340	96.7407
PFD, $m = m_2$	83.7138	97.0556	97.0820	89.8632	95.8340	96.7407
PFD, $m = m_3$	83.6626	97.0000	97.0557	90.0241	95.8340	96.7176
RP, $m = m_1$	83.4374	96.9444	96.5037	88.8174	95.7188	96.7499
RP, $m = m_2$	**83.9492**	96.2778	96.8980	89.7023	95.6694	96.7499
RP, $m = m_3$	83.7650	96.7222	97.0557	89.4610	95.7846	96.7315
Oja, $m = m_1$	83.6831	96.3889	96.6351	89.0587	95.7846	**96.7776**
Oja, $m = m_2$	83.6319	96.9444	96.8980	89.1392	95.7846	**96.7776**
Oja, $m = m_3$	83.5091	**97.1111**	96.8980	89.1392	95.7846	**96.7776**
RFD, $m = m_1$	83.6319	96.2222	96.6877	89.7828	95.8340	95.7666
RFD, $m = m_2$	83.6831	96.4444	96.9243	89.8632	95.8834	96.1267
RFD, $m = m_3$	83.9390	96.9444	97.0820	**90.3459**	96.1139	96.4037
RFD$_0$, $m = m_1$	83.2429	95.9444	96.6877	87.6911	96.3280	96.1636
RFD$_0$, $m = m_2$	83.2634	96.2778	96.9243	88.8174	96.3115	96.3806
RFD$_0$, $m = m_3$	83.2736	96.8889	**97.1083**	88.8978	**96.4762**	96.5560

Table 2: We list the accuracy (%) on test set at the end of one pass with best choice of α_0. The sketch size is set $(m_1, m_2, m_3) = (5, 10, 20)$ for “a9a”, “gisette”, “sido0” and $(m_1, m_2, m_3) = (20, 30, 50)$ for “farm-ads”, “rcv1” and “real-sim”.

7. Conclusions

In this paper we have proposed a novel sketching method robust frequent directions (RFD), and our theoretical analysis shows that RFD is superior to FD. We have also studied the use of RFD in the second order online learning algorithms. The online learning algorithm with RFD achieves better performance than baselines. It is worth pointing out that the application of RFD is not limited to convex online optimization. In future work, we would like to explore the use of RFD in stochastic optimization and non-convex problems.

Acknowledgments

We thank the anonymous reviewers for their helpful suggestions. Luo Luo, Cheng Chen and Zhihua Zhang have been supported by the National Natural Science Foundation of China (No. 61572017 and 11771002) and by Beijing Municipal Commission of Science and Technology under Grant No. 181100008918005. Wu-Jun Li has been supported by the NSFC-NRF Joint Research Project (No. 61861146001) and by the NSFC (No. 61472182).

Algorithms	a9a	gisette	sido0	farm-ads	rcv1	real-sim
ADAGRAD	9.8279e-05	1.9677e-04	1.9504e-04	3.1976e-04	7.0888e-04	3.2917e-04
FULL-ON	2.3141e-04	2.6296e-01	1.6299e-01	/	/	/
FD, $m = m_1$	2.8552e-04	5.2073e-04	6.0705e-04	1.6387e-03	2.0727e-02	6.6130e-03
FD, $m = m_2$	2.7276e-04	7.2830e-04	7.8723e-04	3.2000e-03	4.1148e-02	1.3530e-02
FD, $m = m_3$	3.4899e-04	3.4821e-03	1.4404e-03	5.1365e-03	7.2211e-02	3.0713e-02
PFD, $m = m_1$	2.6090e-04	5.8330e-04	5.7862e-04	2.4524e-02	2.0121e-02	6.6757e-03
PFD, $m = m_2$	2.8206e-04	2.0234e-03	7.7824e-04	3.2372e-02	4.1385e-02	1.2965e-02
PFD, $m = m_3$	3.2096e-04	3.3901e-03	1.4680e-03	5.4829e-02	7.1159e-02	3.0677e-02
RP, $m = m_1$	1.3597e-04	3.2097e-04	3.6333e-04	7.3933e-04	1.8736e-03	1.5551e-03
RP, $m = m_2$	2.7308e-04	7.2830e-04	7.8723e-04	3.2000e-03	2.8015e-03	1.8377e-03
RP, $m = m_3$	3.3307e-04	3.4821e-03	1.4404e-03	5.1365e-03	4.1585e-03	2.1537e-03
Oja, $m = m_1$	1.5500e-04	6.0334e-04	2.9098e-04	1.3061e-03	6.7158e-03	5.2530e-03
Oja, $m = m_2$	1.6719e-04	7.3481e-04	2.4472e-04	2.8887e-03	4.1148e-02	1.3530e-02
Oja, $m = m_3$	1.6631e-04	3.3918e-03	1.3920e-03	4.4386e-03	7.2211e-02	3.0713e-02
RFD, $m = m_1$	2.8549e-04	5.1545e-04	6.0296e-04	2.0484e-03	2.0557e-02	9.7858e-03
RFD, $m = m_2$	3.1813e-04	7.5013e-04	7.5699e-04	3.8129e-03	4.0695e-02	1.6527e-02
RFD, $m = m_3$	3.3405e-04	3.3495e-03	1.4472e-03	5.2458e-03	7.1764e-02	3.0175e-02
RFD$_0$, $m = m_1$	2.7466e-04	6.8607e-04	5.9339e-04	2.0843e-03	2.0033e-02	9.8373e-03
RFD$_0$, $m = m_2$	1.9542e-04	8.0961e-04	7.6749e-04	3.7857e-03	4.0779e-02	1.6892e-02
RFD$_0$, $m = m_3$	2.3561e-04	1.4328e-03	7.6749e-04	5.5628e-03	7.3725e-02	3.0490e-02

Table 3: We list the average iteration cost corresponding to Table 2.

Appendix A. Accelerating by Doubling Space

The cost of FD (Algorithm 1) is dominated by the steps of SVD. It takes $\mathcal{O}(Tm^2d)$ time by standard SVD in total. We can accelerate FD by doubling the sketch size (Liberty, 2013). The details are shown in Algorithm 4. Then the SVD is called only every m rows of $\mathbf{A}$ come and the time complexity is reduced to $\mathcal{O}(Tmd)$. Similarly, RFD can also be speeded up in this way. We demonstrate it in Algorithm 5.

We can apply similar strategy on RFD-SON, just as Algorithm 6 shows. For $\alpha^{(t)} > 0$, the parameter $\mathbf{w}^{(t)}$ can be updated in $\mathcal{O}(md)$ cost by Woodbury identity (Luo et al., 2016). Suppose $\mathbf{B}^{(t)} \in \mathbb{R}^{m' \times d}$, where $m' \leq 2m$. We have

$$\begin{aligned}
\mathbf{u}^{(t+1)} &= \mathbf{w}^{(t)} - \frac{1}{\alpha^{(t)}}\big(\mathbf{g}^{(t)} - (\mathbf{B}^{(t)})^\top \mathbf{M}^{(t)} \mathbf{B}^{(t)} \mathbf{g}^{(t)}\big), \\
\mathbf{w}^{(t+1)} &= \mathbf{u}^{(t+1)} - \gamma^{(t)}\big(\mathbf{x}^{(t)} - (\mathbf{B}^{(t)})^\top \mathbf{M}^{(t)} \mathbf{B}^{(t)} \mathbf{x}^{(t)}\big),
\end{aligned}$$

where

$$\begin{aligned}
\gamma^{(t)} &= \frac{\tau\big((\mathbf{u}^{(t)})^\top \mathbf{x}^{(t)}\big)}{(\mathbf{x}^{(t)})^\top \mathbf{x}^{(t)} - (\mathbf{x}^{(t)})^\top (\mathbf{B}^{(t)})^\top \mathbf{M}^{(t)} \mathbf{B}^{(t)} \mathbf{x}^{(t)}}, \\
\mathbf{M}^{(t)} &= \big(\mathbf{B}^{(t)} (\mathbf{B}^{(t)})^\top + \alpha^{(t)} \mathbf{I}_m\big)^{-1}, \\
\tau(z) &= \operatorname{sgn}(z) \max\{|z| - 1, 0\}.
\end{aligned}$$

Let's check the cost of the above steps in detail. The matrix $\mathbf{M}^{(t)}$ costs $\mathcal{O}(m^2)$ space and the computation of $\mathbf{M}^{(t)}(\mathbf{B}^{(t)}\mathbf{g}^{(t)})$ or $\mathbf{M}^{(t)}(\mathbf{B}^{(t)}\mathbf{x}^{(t)})$ takes $\mathcal{O}(md + m^3)$ time given $\mathbf{B}^{(t)}(\mathbf{B}^{(t)})^\top +$

Algorithm 4 Fast Frequent Directions

```
Input: $\mathbf{A} = [\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(T)}]^\top \in \mathbb{R}^{T \times d}$, $\mathbf{B}^{(0)} = [\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(m-1)}]^\top$
for $t = m, \dots, T$ do
    $\widehat{\mathbf{B}}^{(t-1)} = \begin{bmatrix} \mathbf{B}^{(t-1)} \\ (\mathbf{a}^{(t)})^\top \end{bmatrix}$
    if $\widehat{\mathbf{B}}^{(t-1)}$ has $2m$ rows
        Compute SVD: $\widehat{\mathbf{B}}^{(t-1)} = \mathbf{U}^{(t-1)} \mathbf{\Sigma}^{(t-1)} (\mathbf{V}^{(t-1)})^\top$
        $\mathbf{B}^{(t)} = \sqrt{\big(\mathbf{\Sigma}_{m-1}^{(t-1)}\big)^2 - \big(\sigma_m^{(t-1)}\big)^2 \mathbf{I}_{m-1}} \cdot \big(\mathbf{V}_{m-1}^{(t-1)}\big)^\top$
    else
        $\mathbf{B}^{(t)} = \widehat{\mathbf{B}}^{(t-1)}$
    end if
end for
Output: $\mathbf{B} = \mathbf{B}^{(T)}$
```

Algorithm 5 Fast Robust Frequent Directions

```
Input: $\mathbf{A} = [\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(T)}]^\top \in \mathbb{R}^{T \times d}$, $\mathbf{B}^{(m-1)} = [\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(m-1)}]^\top$, $\alpha^{(m-1)} = 0$
for $t = m, \dots, T$ do
    $\widehat{\mathbf{B}}^{(t-1)} = \begin{bmatrix} \mathbf{B}^{(t-1)} \\ (\mathbf{a}^{(t)})^\top \end{bmatrix}$
    if $\widehat{\mathbf{B}}^{(t-1)}$ has $2m$ rows
        Compute SVD: $\widehat{\mathbf{B}}^{(t-1)} = \mathbf{U}^{(t-1)} \mathbf{\Sigma}^{(t-1)} (\mathbf{V}^{(t-1)})^\top$
        $\mathbf{B}^{(t)} = \sqrt{\big(\mathbf{\Sigma}_{m-1}^{(t-1)}\big)^2 - \big(\sigma_m^{(t-1)}\big)^2 \mathbf{I}_{m-1}} \cdot \big(\mathbf{V}_{m-1}^{(t-1)}\big)^\top$
        $\alpha^{(t)} = \alpha^{(t-1)} + \big(\sigma_m^{(t-1)}\big)^2 / 2$
    else
        $\mathbf{B}^{(t)} = \widehat{\mathbf{B}}^{(t-1)}$
        $\alpha^{(t)} = \alpha^{(t-1)}$
    end if
end for
Output: $\mathbf{B} = \mathbf{B}^{(T)}$ and $\alpha = \alpha^{(T)}$
```

$\alpha^{(t)} \mathbf{I}_{m'}$. The result of $\mathbf{B}^{(t)} (\mathbf{B}^{(t)})^\top$ also can be obtained in $\mathcal{O}(md)$ time. If we have executed SVD at current iteration (when $\widehat{\mathbf{B}}^{(t-1)}$ has $2m$ rows), $\mathbf{B}^{(t)} (\mathbf{B}^{(t)})$ is diagonal and we can directly obtain the SVD of

$$\mathbf{B}^{(t)}(\mathbf{B}^{(t)})^\top = \big(\mathbf{\Sigma}_{m-1}^{(t-1)}\big)^2 - \big(\sigma_m^{(t-1)}\big)^2\mathbf{I}_{m-1},$$

otherwise it can be updated incrementally in $\mathcal{O}(md)$ as follows

$$\mathbf{B}^{(t)}(\mathbf{B}^{(t)})^\top = \begin{bmatrix} \mathbf{B}^{(t-1)}(\mathbf{B}^{(t-1)})^\top & \sqrt{\mu_t+\eta_t}\mathbf{B}^{(t-1)}\mathbf{g}^{(t)} \\ \sqrt{\mu_t+\eta_t}(\mathbf{B}^{(t-1)}\mathbf{g}^{(t)})^\top & (\mu_t+\eta_t)(\mathbf{g}^{(t)})^\top\mathbf{g}^{(t)} \end{bmatrix}.$$

Since we have $m \ll d$, all above operations only require $\mathcal{O}(md)$ time and space complexity in total for each iteration.

In the case of $\alpha^{(t)} = 0$, we can iterate $\mathbf{u}^{(t)}$ and $\mathbf{w}^{(t)}$ by using SVD on $\mathbf{B}^{(t)}$. Let the condensed SVD of $\mathbf{B}^{(t)}$ be $\mathbf{B}^{(t)} = \mathbf{U\Sigma V}^\top$, where $\mathbf{U} \in \mathbb{R}^{m'\times r}$, $\mathbf{\Sigma} \in \mathbb{R}^{r\times r}$, $\mathbf{V} \in \mathbb{R}^{r\times d}$ and $r = \mathrm{rank}(\mathbf{B}^{(t)}) < m'$. Then we have

$$\begin{aligned} \mathbf{H}^{(t)} &= (\mathbf{B}^{(t)})^\top\mathbf{B}^{(t)} = \mathbf{V\Sigma}^2\mathbf{V}^\top, \\ (\mathbf{H}^{(t)})^\dagger &= \mathbf{V\Sigma}^{-2}\mathbf{V}^\top. \end{aligned}$$

We can update $\mathbf{u}^{(t+1)}$ and $\mathbf{w}^{(t+1)}$ as follows

$$\begin{aligned} \mathbf{u}^{(t+1)} &= \mathbf{w}^{(t)} - \mathbf{V}^\top(\mathbf{\Sigma}^{-2}(\mathbf{V}^\top\mathbf{g}^{(t)})), \\ \mathbf{w}^{(t+1)} &= \operatorname*{argmin}_{\mathbf{w}\in\{\mathbf{w}_1^{(t)},\mathbf{w}_2^{(t)}\}} \|\mathbf{w}-\mathbf{u}^{(t+1)}\|_{\mathbf{H}^{(t)}}, \end{aligned}$$

where

$$\begin{aligned} \mathbf{w}_1^{(t)} =& \mathbf{u}^{(t+1)} - \frac{\tau\big((\mathbf{u}^{(t+1)})^\top\mathbf{x}^{(t)}\big)}{(\mathbf{x}^{(t)})^\top\mathbf{V\Sigma}^{-2}\mathbf{V}^\top\mathbf{x}^{(t)}}\mathbf{V\Sigma}^{-2}\mathbf{V}^\top\mathbf{x}^{(t)}, \\ \mathbf{w}_2^{(t)} =& \mathbf{u}^{(t+1)} - \frac{\tau\big((\mathbf{u}^{(t+1)})^\top\mathbf{x}^{(t)}\big)}{(\mathbf{x}^{(t)})^\top\mathbf{x}^{(t)} - (\mathbf{x}^{(t)})^\top\mathbf{V\Sigma}^{-2}\mathbf{V}^\top\mathbf{x}^{(t)}}(\mathbf{x}^{(t)} - \mathbf{V\Sigma}^{-2}\mathbf{V}^\top\mathbf{x}^{(t)}). \end{aligned}$$

The iteration costs $\mathcal{O}(m^2d)$ in total (dominated by the SVD of $\mathbf{B}^\top$), but $\alpha^{(t)} = 0$ only appears at a few early iterations. Hence the average iteration complexity of RFD-SON is dominated by the case $\alpha^{(t)} > 0$ which takes $\mathcal{O}(md)$. Note that the algorithm is also valid without the smoothness of f_t. We can replace the gradient ∇f_t with the corresponding subgradient.

Appendix B. The Proof of Theorem 1

In this section, we firstly provide several lemmas from the book "Topics in matrix analysis" (Horn and Johnson, 1991), then we prove Theorem 1. The proof of Lemma 1 and 2 can be found in the book and we give the proof of Lemma 3 here.

Lemma 1 (Theorem 3.4.5 of (Horn and Johnson, 1991)) *Let* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$ *be given, and suppose* $\mathbf{A}$*,* $\mathbf{B}$ *and* $\mathbf{A}-\mathbf{B}$ *have decreasingly ordered singular values,* $\sigma_1(\mathbf{A}) \geq \cdots \geq \sigma_q(\mathbf{A})$*,* $\sigma_1(\mathbf{B}) \geq \cdots \geq \sigma_q(\mathbf{B})$*, and* $\sigma_1(\mathbf{A}-\mathbf{B}) \geq \cdots \geq \sigma_q(\mathbf{A}-\mathbf{B})$*, where* $q = \min\{m,n\}$*. Define* $s_i(\mathbf{A},\mathbf{B}) \equiv |\sigma_i(\mathbf{A}) - \sigma_i(\mathbf{B})|$*,* $i = 1,\dots,q$ *and let* $s_{[1]}(\mathbf{A},\mathbf{B}) \geq \cdots \geq s_{[i]}(\mathbf{A},\mathbf{B})$ *denote a decreasingly ordered rearrangement of the values* $s_i(\mathbf{A},\mathbf{B})$*. Then*

$$\sum_{i=1}^k s_{[i]}(\mathbf{A},\mathbf{B}) \leq \sum_{i=1}^k \sigma_i(\mathbf{A}-\mathbf{B})\,\textit{for } k = 1,\dots,q.$$

Algorithm 6 Fast RFD for Online Newton Step

```
Input: $\alpha^{(0)} = \alpha_0$, $m < d$, $\eta_t = \mathcal{O}(1/t)$ and $\mathbf{B}^{(0)} = \mathbf{0}^{m\times d}$.
for $t = 1, \dots, T$ do
    Receive example $\mathbf{x}^{(t)}$, and loss function $f_t(\mathbf{w})$
    Predict the output of $\mathbf{x}^{(t)}$ by $\mathbf{w}^{(t)}$ and suffer loss $f_t(\mathbf{w}^{(t)})$
    $\mathbf{g}^{(t)} = \nabla f_t(\mathbf{w}^{(t)})$
    $\widehat{\mathbf{B}}^{(t-1)} = \begin{bmatrix} \mathbf{B}^{(t-1)} \\ (\sqrt{\mu_t + \eta_t}\mathbf{g}^{(t)})^\top \end{bmatrix}$
    if $\widehat{\mathbf{B}}^{(t-1)}$ has $2m$ rows
        Compute SVD: $\widehat{\mathbf{B}}^{(t-1)} = \mathbf{U}^{(t-1)}\mathbf{\Sigma}^{(t-1)}(\mathbf{V}^{(t-1)})^\top$
        $\mathbf{B}^{(t)} = \sqrt{\big(\mathbf{\Sigma}_{m-1}^{(t-1)}\big)^2 - \big(\sigma_m^{(t-1)}\big)^2\mathbf{I}_{m-1}} \cdot (\mathbf{V}_{m-1}^{(t-1)})^\top$
        $\alpha^{(t)} = \alpha^{(t-1)} + \big(\sigma_m^{(t-1)}\big)^2/2$
    else
        $\mathbf{B}^{(t)} = \widehat{\mathbf{B}}^{(t-1)}$
        $\alpha^{(t)} = \alpha^{(t-1)}$
    end if
    $\mathbf{H}^{(t)} = (\mathbf{B}^{(t)})^\top\mathbf{B}^{(t)} + \alpha^{(t)}\mathbf{I}_d$
    $\mathbf{u}^{(t+1)} = \mathbf{w}^{(t)} - (\mathbf{H}^{(t)})^\dagger\mathbf{g}^{(t)}$
    $\mathbf{w}^{(t+1)} = \mathrm{argmin}_{\mathbf{w}\in\mathcal{K}_t} \|\mathbf{w} - \mathbf{u}^{(t+1)}\|_{\mathbf{H}^{(t)}}$
end for
```

Lemma 2 (Corollary 3.5.9 of (Horn and Johnson, 1991)) *Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$ be given, and let $q = \min\{m, n\}$. The following are equivalent*

1. $|\!|\!|\mathbf{A}|\!|\!| \leq |\!|\!|\mathbf{B}|\!|\!|$ *for every unitarily invariant norm* $|\!|\!|\cdot|\!|\!|$ *on* $\mathbb{R}^{m\times n}$.
2. $N_k(\mathbf{A}) \leq N_k(\mathbf{B})$ *for* $k = 1, \dots, q$ *where* $N_k(\mathbf{X}) \equiv \sum_{i=1}^k \sigma_k(\mathbf{X})$ *denotes Ky Fan* k*-norm.*

Lemma 3 (Page 215 of (Horn and Johnson, 1991)) *Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$ be given, and let $q = \min\{m, n\}$. Define the diagonal matrix $\mathbf{\Sigma}(\mathbf{A}) = [\sigma_{ij}] \in \mathbb{R}^{m\times n}$ by $\sigma_{ii} = \sigma_i(\mathbf{A})$, all other $\sigma_{ij} = 0$, where $\sigma_1(\mathbf{A}) \geq, \dots, \geq \sigma_q(\mathbf{A})$ are the decreasingly ordered singular values of $\mathbf{A}$. We define $\mathbf{\Sigma}(\mathbf{B})$ similarly. Then we have $|\!|\!|\mathbf{A} - \mathbf{B}|\!|\!| \geq |\!|\!|\mathbf{\Sigma}(\mathbf{A}) - \mathbf{\Sigma}(\mathbf{B})|\!|\!|$ for every unitarily invariant norm $|\!|\!|\cdot|\!|\!|$.*

Proof Using the notation of Lemma 1 and 2, matrices $\mathbf{A}-\mathbf{B}$ and $\mathbf{\Sigma}(\mathbf{A})-\mathbf{\Sigma}(\mathbf{B})$ have the decreasingly ordered singular values $\sigma_1(\mathbf{A} - \mathbf{B}) \geq \dots \geq \sigma_q(\mathbf{A} - \mathbf{B})$ and $s_{[1]}(\mathbf{A}, \mathbf{B}) \geq \dots \geq s_{[q]}(\mathbf{A}, \mathbf{B})$. Then we have

$$N_k(\mathbf{A} - \mathbf{B}) = \sum_{i=1}^{k} \sigma_i(\mathbf{A} - \mathbf{B}) \geq \sum_{i=1}^{k} s_{[i]}(\mathbf{A}, \mathbf{B}) = N_k(\mathbf{\Sigma}(\mathbf{A}) - \mathbf{\Sigma}(\mathbf{B})), \tag{15}$$

where the inequality is obtained by Lemma 1. The Lemma 2 implies (15) is equivalent to $\|\|\mathbf{A} - \mathbf{B}\|\| \geq \|\|\mathbf{\Sigma}(\mathbf{A}) - \mathbf{\Sigma}(\mathbf{B})\|\|$ for every unitarily invariant norm $\|\| \cdot \|\|$. ■

Then we give the proof of Theorem 1 as follows:
Proof Using the notation in above lemmas, we can bound the objective function as follows

$$\begin{aligned}\left\|\mathbf{M} - \mathbf{C}\mathbf{C}^\top - \delta\mathbf{I}_d\right\|_2 &\geq \left\|\mathbf{\Sigma}(\mathbf{M}) - \mathbf{\Sigma}(\mathbf{C}\mathbf{C}^\top + \delta\mathbf{I}_d)\right\|_2 \\ &= \max_{i\in\{1,\dots,d\}} \left|\sigma_i(\mathbf{M}) - \sigma_i(\mathbf{C}\mathbf{C}^\top) - \delta\right| \\ &\geq \max_{i\in\{k+1,\dots,d\}} \left|\sigma_i(\mathbf{M}) - \sigma_i(\mathbf{C}\mathbf{C}^\top) - \delta\right| \\ &= \max_{i\in\{k+1,\dots,d\}} \left|\sigma_i(\mathbf{M}) - \delta\right| \\ &\geq \max_{i\in\{k+1,\dots,d\}} \left|\sigma_i(\mathbf{M}) - \widehat{\delta}\right|.\end{aligned}$$

The first inequality is obtained by Lemma 3 since the spectral norm is unitarily invariant, and the second inequality is the property of maximization operator. The last inequality can be checked easily by the property of max operation and the equivalence of SVD and eigenvector decomposition for positive semi-definite matrix. The first equality is based on the definition of spectral norm. The second equality holds due to the fact $\mathrm{rank}(\mathbf{C}\mathbf{C}^\top) \leq k$ which leads $\sigma_i(\mathbf{C}\mathbf{C}^\top) = 0$ for any $i > k$. Note that all above equalities occur for $\mathbf{C} = \widehat{\mathbf{C}} = \mathbf{U}_k(\mathbf{\Sigma}_k - \xi\mathbf{I}_k)^{1/2}\mathbf{Q}$, $\delta = \widehat{\delta}$ and $\xi \in [\sigma_d, \sigma_{k+1}]$. Hence we prove the optimality of $(\widehat{\mathbf{C}}, \widehat{\delta})$.

The approximation error of rank-k SVD corresponds to the objective of (8) by taking $\mathbf{C} = \mathbf{U}_k(\mathbf{\Sigma}_k)^{1/2}$ and $\delta = 0$, which is impossible to be smaller than the minimum. It is easy to verify we have $\widehat{\mathbf{C}} = \mathbf{U}_k(\mathbf{\Sigma}_k)^{1/2}$ and $\widehat{\delta} = 0$ if and only of $\mathrm{rank}(\mathbf{M}) \leq k$. ■

Theorem 1 means the choice of ξ in the solution of problem (9) is not unique, but taking $\xi = \sigma_{k+1}$ minimizes the condition number of $\widehat{\mathbf{C}}\widehat{\mathbf{C}}^\top + \widehat{\delta}\mathbf{I}_d$. Hence, we use $\xi = \sigma_{k+1}$ in the derivation of RFD. We also demonstrate similar result with respect to Frobenius norm in Corollary 1. This analysis includes the global optimality of the problem, while Zhang (2014)'s analysis only prove the solution is locally optimal.

Corollary 1 *Using the same notation in Theorem 1, the pair* $(\tilde{\mathbf{C}}, \tilde{\delta})$ *defined as*

$$\tilde{\mathbf{C}} = \mathbf{U}_k(\mathbf{\Sigma}_k - \tilde{\delta}\mathbf{I}_k)^{1/2}\mathbf{V} \quad \text{and} \quad \tilde{\delta} = \frac{1}{d-k}\sum_{i=j+1}^{d}\sigma_i$$

is the global minimizer of

$$\min_{\mathbf{C}\in\mathbb{R}^{d\times k}, \delta\in\mathbb{R}} \|\mathbf{M} - \mathbf{C}\mathbf{C}^\top - \delta\mathbf{I}_d\|_F^2,$$

where $\mathbf{V}$ *is an arbitrary* $k \times k$ *orthogonal matrix.*

Proof We have the result similar to Theorem 1.

$$\begin{aligned}\|\mathbf{M} - \mathbf{C}\mathbf{C}^\top - \delta\mathbf{I}_d\|_F^2 \geq& \|\mathbf{\Sigma}(\mathbf{M}) - \mathbf{\Sigma}(\mathbf{C}\mathbf{C}^\top + \delta\mathbf{I}_d)\|_F^2 \\ =& \sum_{i=1}^{d}(\sigma_i(\mathbf{M}) - \sigma_i(\mathbf{C}\mathbf{C}^\top) - \delta)^2 \\ \geq& \sum_{i=k+1}^{d}(\sigma_i(\mathbf{M}) - \sigma_i(\mathbf{C}\mathbf{C}^\top) - \delta)^2 \\ =& \sum_{i=k+1}^{d}(\sigma_i(\mathbf{M}) - \delta)^2 \\ \geq& \sum_{i=k+1}^{d}(\sigma_i(\mathbf{M}) - \tilde{\delta})^2\end{aligned}$$

The first four steps are similar to the ones of Theorem 1, but replace the spectral norm and absolute operator with Frobenius norm and square function. The last step comes from the property of the mean value.

We can check that all above equalities occur for $\mathbf{C} = \tilde{\mathbf{C}}$ and $\delta = \tilde{\delta}$, which completes the proof. ■

Appendix C. The Proof of Theorem 2

Proof The Algorithm 2 implies the singular values of $(\widehat{\mathbf{B}}^{(t-1)})^\top\widehat{\mathbf{B}}^{(t-1)} + \alpha^{(t-1)}\mathbf{I}$ are

$$(\sigma_1^{(t-1)})^2 + \alpha^{(t-1)} \geq \cdots \geq (\sigma_m^{(t-1)})^2 + \alpha^{(t-1)} \geq \alpha^{(t-1)} = \cdots = \alpha^{(t-1)}.$$

Then we can use Theorem 1 by taking

$$\begin{aligned}\mathbf{M} =&(\widehat{\mathbf{B}}^{(t-1)})^\top\widehat{\mathbf{B}}^{(t-1)} + \alpha^{(t-1)}\mathbf{I}, \\ k =&m-1, \\ \xi =&\sigma_{k+1} = (\sigma_m^{(t-1)})^2 + \alpha^{(t-1)} \\ \widehat{\mathbf{C}} =&\mathbf{V}_{m-1}^{(t-1)}\sqrt{\big(\mathbf{\Sigma}_{m-1}^{(t-1)}\big)^2 - \big(\sigma_m^{(t-1)}\big)^2\mathbf{I}_{m-1}} = (\mathbf{B}^{(t)})^\top, \\ \widehat{\delta} =&[(\sigma_m^{(t-1)})^2 + \alpha^{(t-1)} + \alpha^{(t-1)}]/2 = \alpha^{(t)},\end{aligned}$$

which just means that $(\mathbf{B}^{(t)}, \alpha^{(t)})$ is the minimizer of the problem in this theorem. ■

Appendix D. The Proof of Theorem 3

The algorithms of FD and RFD share the same $\mathbf{B}^{(t)}$ and we have Lemma 4 (Ghashami et al., 2016) as follows.

Lemma 4 *For any $k < m$ and using the notation of Algorithm 1 or Algorithm 2, we have*

$$\mathbf{A}^\top \mathbf{A} - \mathbf{B}^\top \mathbf{B} \succeq \mathbf{0}, \tag{16}$$

$$\sum_{t=1}^{T-1} (\sigma_m^{(t)})^2 \leq \frac{1}{m-k} \|\mathbf{A} - [\mathbf{A}]_k\|_F^2. \tag{17}$$

Then we prove the Theorem 3 based on Lemma 4.

Proof Define $(\mathbf{B}^{(0)})^\top \mathbf{B}^{(0)} = \mathbf{0}^{d\times d}$, then we can derive the error bound as follows

$$\begin{aligned}
&\left\|\mathbf{A}^\top \mathbf{A} - (\mathbf{B}^\top \mathbf{B} + \alpha \mathbf{I}_d)\right\|_2 \\
=&\left\| \sum_{t=1}^{T} \left[(\mathbf{a}^{(t)})^\top \mathbf{a}^{(t)} - (\mathbf{B}^{(t)})^\top \mathbf{B}^{(t)} + (\mathbf{B}^{(t-1)})^\top \mathbf{B}^{(t-1)} - \frac{1}{2}(\sigma_m^{(t-1)})^2 \mathbf{I}_d \right] \right\|_2 \\
\leq& \sum_{t=1}^{T} \left\| (\mathbf{a}^{(t)})^\top \mathbf{a}^{(t)} - (\mathbf{B}^{(t)})^\top \mathbf{B}^{(t)} + (\mathbf{B}^{(t-1)})^\top \mathbf{B}^{(t-1)} - \frac{1}{2}(\sigma_m^{(t-1)})^2 \mathbf{I}_d \right\|_2 \\
=& \sum_{t=1}^{T} \left\| \mathbf{V}_{m-1}^{(t-1)} (\mathbf{\Sigma}_{m-1}^{(t-1)})^2 (\mathbf{V}_{m-1}^{(t-1)})^\top - \mathbf{V}_{m-1}^{(t-1)} [(\mathbf{\Sigma}_{m-1}^{(t-1)})^2 - (\sigma_m^{(t-1)})^2 \mathbf{I}_d] (\mathbf{V}_{m-1}^{(t-1)})^\top - \frac{1}{2}(\sigma_m^{(t-1)})^2 \mathbf{I}_d \right\|_2 \\
=& \sum_{t=1}^{T} \left\| (\sigma_m^{(t-1)})^2 \mathbf{V}_m^{(t-1)} (\mathbf{V}_m^{(t-1)})^\top - \frac{1}{2}(\sigma_m^{(t-1)})^2 \mathbf{I}_d \right\|_2 \\
=& \frac{1}{2} \sum_{t=1}^{T-1} (\sigma_m^{(t)})^2 \\
\leq& \frac{1}{2(m-k)} \|\mathbf{A} - [\mathbf{A}]_k\|_F^2.
\end{aligned}$$

The first three equalities are direct from the procedure of the algorithm, and the last one is based on the fact that $\mathbf{V}^{(t-1)}$ is column orthonormal. The first inequality comes from the triangle inequality of spectral norm. The last one can be obtained by the result (17) of Lemma 4. ∎

We also have similar error bound for fast RFD with doubling space. We first rewrite Algorithm 5 as the block formulation. Consider the procedure of Algorithm 5, we suppose that matrix $\widehat{\mathbf{B}}^{(t-1)}$ has $2m$ rows at round $t = p_1, p_2, \ldots, p_{T'-1}$, where $1 < p_1 < p_2 < \cdots < p_{T'-1} \leq T$. Letting $p_0 = 0$ and $p_{T'} = T$, we can partition matrix $\mathbf{A}$ into T' blocks

$$\mathbf{A} = \left[\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \cdots, \mathbf{A}^{(T')}\right]^\top, \tag{18}$$

where

$$\mathbf{A}^{(t')} = \left[\mathbf{a}^{(p_{t'-1}+1)}, \mathbf{a}^{(p_{t'-1}+2)}, \cdots, \mathbf{a}^{(p_{t'})}\right]^\top \text{ for } t' = 1, 2, \ldots, T'. \tag{19}$$

Based on the notation of (18) and (19), we can rewrite Algorithm 5 as Algorithm 7. It is obvious that two algorithms have the same output results. We present Lemma 5 which extends Lemma 4 to block version and establishes the error bound for Algorithm 7 in Corollary 2.

Algorithm 7 Fast Robust Frequent Directions (Block Formulation)

```
Input: $\mathbf{A} = [\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(T')}]^\top \in \mathbb{R}^{T \times d}$, $\mathbf{B}^{(1)} = (\mathbf{A}^{(1)})^\top$, $\alpha^{(1)} = 0$
for $t' = 2, \dots, T'$ do
    $\widehat{\mathbf{B}}^{(t-1)} = \begin{bmatrix} \mathbf{B}^{(t'-1)} \\ (\mathbf{A}^{(t')})^\top \end{bmatrix}$
    Compute SVD: $\widehat{\mathbf{B}}^{(t'-1)} = \mathbf{U}^{(t'-1)} \mathbf{\Sigma}^{(t'-1)} (\mathbf{V}^{(t'-1)})^\top$
    $\mathbf{B}^{(t')} = \sqrt{\big(\mathbf{\Sigma}_{m-1}^{(t'-1)}\big)^2 - \big(\sigma_m^{(t'-1)}\big)^2 \mathbf{I}_{m-1}} \cdot \big(\mathbf{V}_{m-1}^{(t'-1)}\big)^\top$
    $\alpha^{(t')} = \alpha^{(t'-1)} + \big(\sigma_m^{(t'-1)}\big)^2/2$
end for
Output: $\mathbf{B} = \mathbf{B}^{(T')}$ and $\alpha = \alpha^{(T')}$
```

Lemma 5 *For any $k < m$ and using the notation of Algorithm 7, we have*

$$\sum_{t'=1}^{T'-1} (\sigma_m^{(t')})^2 \leq \frac{1}{m-k} \|\mathbf{A} - [\mathbf{A}]_k\|_F^2.$$

Proof We let $\sigma_m^{(T')} = 0$. For any unit vector $\mathbf{x} \in \mathbb{R}^d$, the procedure of Algorithm 7 implies

$$\begin{aligned} & \|\mathbf{A}\mathbf{x}\|^2 - \|\mathbf{B}\mathbf{x}\|^2 \\ = & \sum_{t'=1}^{T'} \Big(\|\mathbf{A}^{(t')}\mathbf{x}\|^2 + \|\mathbf{B}^{(t'-1)}\mathbf{x}\|^2 - \|\mathbf{B}^{(t')}\mathbf{x}\|^2 \Big) \\ = & \sum_{t'=1}^{T'} \mathbf{x}^\top \Big((\mathbf{A}^{(t')})^\top \mathbf{A}^{(t')} + (\mathbf{B}^{(t'-1)})^\top (\mathbf{B}^{(t'-1)}) - (\mathbf{B}^{(t')})^\top \mathbf{B}^{(t')} \Big) \mathbf{x} \\ = & \sum_{t'=1}^{T'} \mathbf{x}^\top \Big((\widehat{\mathbf{B}}^{(t')})^\top \widehat{\mathbf{B}}^{(t')} - (\mathbf{B}^{(t')})^\top \mathbf{B}^{(t')} \Big) \mathbf{x} \\ \leq & \sum_{t'=1}^{T'} \big(\sigma_m^{(t')}\big)^2. \end{aligned} \tag{20}$$

Using the property of Frobenius norm, we have

$$\begin{aligned} \|\widehat{\mathbf{B}}^{(t'-1)}\|_F^2 = & \|\mathbf{\Sigma}^{(t'-1)}\|_F^2 \\ \geq & \Big\| \sqrt{\big(\mathbf{\Sigma}_m^{(t'-1)}\big)^2 - \big(\sigma_m^{(t'-1)}\big)^2 \mathbf{I}_m} \Big\|_F^2 + m(\sigma_m^{(t'-1)})^2 \\ = & \|\mathbf{B}^{(t')}\|_F^2 + m(\sigma_m^{(t'-1)})^2. \end{aligned} \tag{21}$$

The term $\|\mathbf{A}\|_F^2$ satisfies

$$\|\mathbf{A}\|_F^2 = \sum_{t'=1}^{T'} \|\mathbf{A}^{(t')}\|_F^2$$

$$
\begin{aligned}
&=\sum_{t'=1}^{T'}\left(\|\widehat{\mathbf{B}}^{(t'-1)}\|_F^2-\|\mathbf{B}^{(t'-1)}\|_F^2\right)\\
&\geq\sum_{t'=1}^{T'}\left(\|\mathbf{B}^{(t')}\|_F^2+m(\sigma_m^{(t'-1)})^2-\|\mathbf{B}^{(t'-1)}\|_F^2\right)\\
&=\|\mathbf{B}\|_F^2+m\sum_{t'=1}^{T'}\left(\sigma_m^{(t')}\right)^2, \qquad (22)
\end{aligned}
$$

where the inequality is due to (21).

Let $\mathbf{y}_i$ be the singular vectors of $\mathbf{A}$ with respect to $\sigma_i(\mathbf{A})$. Then we have

$$
\begin{aligned}
m\sum_{t'=1}^{T'}(\sigma_m^{(t'-1)})^2 \leq&\|\mathbf{A}\|_F^2-\|\mathbf{B}\|_F^2\\
=&\sum_{i=1}^{k}\|\mathbf{A}\mathbf{y}_i\|_F^2+\sum_{i=k+1}^{d}\|\mathbf{A}\mathbf{y}_i\|_F^2-\|\mathbf{B}\|_F^2\\
=&\sum_{i=1}^{k}\|\mathbf{A}\mathbf{y}_i\|_F^2+\|\mathbf{A}-[\mathbf{A}]_k\|_F^2-\|\mathbf{B}\|_F^2\\
\leq&\|\mathbf{A}-[\mathbf{A}]_k\|_F^2+\sum_{i=1}^{k}\left(\|\mathbf{A}\mathbf{y}_i\|_F^2-\|\mathbf{B}\mathbf{y}_i\|_F^2\right)\\
\leq&\|\mathbf{A}-[\mathbf{A}]_k\|_F^2+k\sum_{t'=1}^{T'}\left(\sigma_m^{(t')}\right)^2, \qquad (23)
\end{aligned}
$$

where the first inequality comes from (22), the second inequality is based on the fact $\sum_{i=1}^{k}\|\mathbf{B}\mathbf{y}_i\|^2 \leq \|\mathbf{B}\|_F^2$, and the last one comes from (20). We can obtain the result of this lemma by (23) directly. ■

Corollary 2 *For any $k < m$ and using the notation of Algorithm 7, we have*

$$\left\|\mathbf{A}^\top\mathbf{A}-(\mathbf{B}^\top\mathbf{B}+\alpha\mathbf{I}_d)\right\|_2 \leq \frac{1}{2(m-k)}\|\mathbf{A}-[\mathbf{A}]_k\|_F^2,$$

where $[\mathbf{A}]_k$ is the best rank-k approximation to $\mathbf{A}$ in both the Frobenius and spectral norms.

Proof Define $(\mathbf{B}^{(0)})^\top\mathbf{B}^{(0)}=\mathbf{0}^{d\times d}$, then we can derive the error bound as follows

$$
\begin{aligned}
&\left\|\mathbf{A}^\top\mathbf{A}-(\mathbf{B}^\top\mathbf{B}+\alpha\mathbf{I}_d)\right\|_2\\
=&\left\|\sum_{t'=1}^{T'}\left[(\mathbf{A}^{(t')})^\top\mathbf{A}^{(t')}-(\mathbf{B}^{(t')})^\top\mathbf{B}^{(t')}+(\mathbf{B}^{(t'-1)})^\top\mathbf{B}^{(t'-1)}-\frac{1}{2}(\sigma_m^{(t'-1)})^2\mathbf{I}_d\right]\right\|_2\\
\leq&\sum_{t'=1}^{T'}\left\|(\mathbf{A}^{(t')})^\top\mathbf{A}^{(t')}-(\mathbf{B}^{(t')})^\top\mathbf{B}^{(t')}+(\mathbf{B}^{(t'-1)})^\top\mathbf{B}^{(t'-1)}-\frac{1}{2}(\sigma_m^{(t'-1)})^2\mathbf{I}_d\right\|_2
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{t'=1}^{T'} \left\| \mathbf{V}_{m-1}^{(t'-1)} (\mathbf{\Sigma}_{m-1}^{(t'-1)})^2 (\mathbf{V}_{m-1}^{(t'-1)})^\top - \mathbf{V}_{m-1}^{(t'-1)} [(\mathbf{\Sigma}_{m-1}^{(t'-1)})^2 - (\sigma_m^{(t'-1)})^2 \mathbf{I}_d] (\mathbf{V}_{m-1}^{(t'-1)})^\top - \frac{1}{2} (\sigma_m^{(t'-1)})^2 \mathbf{I}_d \right\|_2 \\
&= \sum_{t'=1}^{T'} \left\| (\sigma_m^{(t'-1)})^2 \mathbf{V}_m^{(t'-1)} (\mathbf{V}_m^{(t'-1)})^\top - \frac{1}{2} (\sigma_m^{(t'-1)})^2 \mathbf{I}_d \right\|_2 \\
&= \frac{1}{2} \sum_{t'=1}^{T'-1} (\sigma_m^{(t')})^2 \\
&\leq \frac{1}{2(m-k)} \|\mathbf{A} - [\mathbf{A}]_k\|_F^2 .
\end{aligned}
$$

The last inequality is based on Lemma 5 and other steps are similar to the proof of Theorem 3. ■

Appendix E. The Proof of Theorem 4

Proof We can compare $\kappa(\mathbf{M}_{\text{RFD}})$ and $\kappa(\mathbf{M}_{\text{FD}})$ by the fact $\alpha \geq \alpha^{(0)}$ as follows

$$
\begin{aligned}
\kappa(\mathbf{M}_{\text{RFD}}) &= \frac{\sigma_{\max}(\mathbf{B}^\top \mathbf{B}) + \alpha}{\alpha} \\
&\leq \frac{\sigma_{\max}(\mathbf{B}^\top \mathbf{B}) + \alpha_0}{\alpha_0} \\
&= \kappa(\mathbf{M}_{\text{FD}}).
\end{aligned}
$$

The other inequality can be derived as

$$
\begin{aligned}
\kappa(\mathbf{M}_{\text{RFD}}) &= \frac{\sigma_{\max}(\mathbf{B}^\top \mathbf{B}) + \alpha}{\alpha} \\
&\leq \frac{\sigma_{\max}(\mathbf{A}^\top \mathbf{A}) + \alpha}{\alpha} \\
&\leq \frac{\sigma_{\max}(\mathbf{A}^\top \mathbf{A}) + \alpha_0}{\alpha_0} \\
&= \kappa(\mathbf{M}),
\end{aligned}
$$

where the first inequality comes from (16) of Lemma 4 and the others are easy to obtain. ■

Appendix F. The Greedy Low-rank Approximation

We present the greedy low-rank approximation (Brand, 2002; Hall et al., 1998; Levey and Lindenbaum, 2000; Ross et al., 2008) as Algorithm 8. The algorithm does not work in general although $(\mathbf{B}'^{(t)})^\top \mathbf{B}'^{(t)}$ is the best low-rank approximation to $(\widehat{\mathbf{B}}^{(t)})^\top \widehat{\mathbf{B}}^{(t)}$.

We provide an example to show the failure of this method. We define $\tilde{\mathbf{A}} = [\tilde{\mathbf{A}}_1^\top, \tilde{\mathbf{A}}_2^\top]^\top \in \mathbb{R}^{(m-s+1)\times d}$, where $\tilde{\mathbf{A}}_1 \in \mathbb{R}^{(m-1)\times d}$, $\tilde{\mathbf{A}}_2 \in \mathbb{R}^{s\times d}$ and $m \ll s$. Suppose that the smallest singular value of $\tilde{\mathbf{A}}_1$ is λ, and each row of $\tilde{\mathbf{A}}_2$ is $\mathbf{a} \in \mathbb{R}^d$ that satisfies $\|\mathbf{a}\| = \lambda - \epsilon$ and $\tilde{\mathbf{A}}_1^\top \mathbf{a} = \mathbf{0}$, where ϵ is a very small positive number. Since s is much larger than m, a good approximation to $\tilde{\mathbf{A}}^\top \tilde{\mathbf{A}}$ is

Algorithm 8 Greedy Low-rank Approximation

```
Input: $\mathbf{A} = [\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(T)}]^\top \in \mathbb{R}^{T \times d}$, $\mathbf{B}'^{(m-1)} = [\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(m-1)}]^\top$
for $t = m, \ldots, T$ do
    $\widehat{\mathbf{B}}^{(t-1)} = \begin{bmatrix} \mathbf{B}'^{(t-1)} \\ (\mathbf{a}^{(t)})^\top \end{bmatrix}$
    Compute SVD: $\widehat{\mathbf{B}}^{(t-1)} = \mathbf{U}^{(t-1)} \mathbf{\Sigma}^{(t-1)} (\mathbf{V}^{(t-1)})^\top$
    $\mathbf{B}'^{(t)} = \mathbf{\Sigma}_{m-1}^{(t-1)} \big(\mathbf{V}_{m-1}^{(t-1)}\big)^\top$
end for
Output: $\mathbf{B}' = \mathbf{B}'^{(T)}$
```

dominated by $\tilde{\mathbf{A}}_2^\top \tilde{\mathbf{A}}_2$. If we use Algorithm 8 with $\mathbf{A} = \tilde{\mathbf{A}}$, any row of $\tilde{\mathbf{A}}_2$ will be neglected because the m-th singular value of $\widehat{\mathbf{B}}^{(t-1)}$ is $\|\mathbf{a}\| = \lambda - \epsilon < \lambda$, which leads to the fact that output is $\mathbf{B}' = \tilde{\mathbf{A}}_1$. Apparently, $\tilde{\mathbf{A}}_1^\top \tilde{\mathbf{A}}_1$ is not a good approximation to $\mathbf{A}^\top \mathbf{A}$. Hence the shrinking of FD or RFD is necessary. In this example, it reduces the impact of $\tilde{\mathbf{A}}_1$ and let $\tilde{\mathbf{A}}_2$ be involved in final result.

Besides above discussion, Desai et al. (2016) has shown that the greedy algorithm is much worse than FD based methods on data sets “Adversarial” and “ConnectUS”.

Appendix G. The Proof of Theorem 5

Lemma 6 shows the general regret bound for any choice of $\mathbf{H}^{(t)} \succ \mathbf{0}$ in update (2)

$$\begin{aligned} \mathbf{u}^{(t+1)} &= \mathbf{w}^{(t)} - \beta_t (\mathbf{H}^{(t)})^{-1} \mathbf{g}^{(t)}, \\ \mathbf{w}^{(t+1)} &= \operatorname*{argmin}_{\mathbf{w} \in \mathcal{K}_{t+1}} \|\mathbf{w} - \mathbf{u}^{(t+1)}\|_{\mathbf{H}^{(t)}}. \end{aligned}$$

Lemma 6 (Proposition 1 of Luo et al. (2016)) *For any sequence of positive definite matrices* $\mathbf{H}^{(t)}$ *and sequences of losses satisfying Assumption 1 and 2, regret of updates (2) satisfies*

$$2R_T(\mathbf{w}) \leq \|\mathbf{w}\|_{\mathbf{H}^{(0)}}^2 + R_G + R_D,$$

where

$$\begin{aligned} R_G &= \sum_{t=1}^{T} (\mathbf{g}^{(t)})^\top (\mathbf{H}^{(t)})^{-1} \mathbf{g}^{(t)}, \\ R_D &= \sum_{t=1}^{T} (\mathbf{w}^{(t)} - \mathbf{w})^\top (\mathbf{H}^{(t)} - \mathbf{H}^{(t-1)} - \mu^{(t)} \mathbf{g}^{(t)} (\mathbf{g}^{(t)})^\top)(\mathbf{w}^{(t)} - \mathbf{w}). \end{aligned}$$

Then we prove the regret bound for RFD-SON based on Lemma 6 and property of RFD.

Proof Let $\mathbf{V}_\perp^{(t)}$ be the orthogonal complement of $\mathbf{V}_{m-1}^{(t)}$’s column space, that is $\mathbf{V}_{m-1}^{(t)} (\mathbf{V}_{m-1}^{(t)})^\top + \mathbf{V}_\perp^{(t)} (\mathbf{V}_\perp^{(t)})^\top = \mathbf{I}_d$, then we have

$$\mathbf{H}^{(t)} - \mathbf{H}^{(t-1)}$$

$$
\begin{aligned}
=&\alpha^{(t)}\mathbf{I} + (\mathbf{B}^{(t)})^\top \mathbf{B}^{(t)} - \alpha^{(t-1)}\mathbf{I} - (\mathbf{B}^{(t-1)})^\top \mathbf{B}^{(t-1)} \\
=&\frac{1}{2}(\sigma_m^{(t-1)})^2\mathbf{I} - (\sigma_m^{(t-1)})^2 \mathbf{V}_{m-1}^{(t-1)}(\mathbf{V}_{m-1}^{(t-1)})^\top + (\mu_t + \eta_t)\mathbf{g}^{(t)}(\mathbf{g}^{(t)})^\top \\
=&\frac{1}{2}(\sigma_m^{(t-1)})^2\big[\mathbf{V}_\perp^{(t-1)}(\mathbf{V}_\perp^{(t-1)})^\top - \mathbf{V}_{m-1}^{(t-1)}(\mathbf{V}_{m-1}^{(t-1)})^\top\big] + (\mu_t + \eta_t)\mathbf{g}^{(t)}(\mathbf{g}^{(t)})^\top. \qquad (24)
\end{aligned}
$$

Since $\mathbf{H}^{(t)}$ is positive semidefinite for any t, we have $(\mathbf{H}^{(t)})^\dagger = (\mathbf{H}^{(t)})^{-1}$. Combining with Lemma 6, we have

$$
2R_T(\mathbf{w}) \le \alpha_0 \|\mathbf{w}\|^2 + R_G + R_D,
$$

where

$$
R_G = \sum_{t=1}^{T} (\mathbf{g}^{(t)})^\top (\mathbf{H}^{(t)})^{-1}\mathbf{g}^{(t)},
$$

and

$$
R_D = \sum_{t=1}^{T} (\mathbf{w}^{(t)} - \mathbf{w})^\top [\mathbf{H}^{(t)} - \mathbf{H}^{(t-1)} - \mu_t \mathbf{g}^{(t)}(\mathbf{g}^{(t)})^\top](\mathbf{w}^{(t)} - \mathbf{w}).
$$

We can bound R_G as follows

$$
\begin{aligned}
&\sum_{t=1}^{T} (\mathbf{g}^{(t)})^\top (\mathbf{H}^{(t)})^{-1}\mathbf{g}^{(t)} \\
=&\sum_{t=1}^{T} \big\langle (\mathbf{H}^{(t)})^{-1}, \mathbf{g}^{(t)}(\mathbf{g}^{(t)})^\top \big\rangle \\
=&\sum_{t=1}^{T} \frac{1}{\mu_t + \eta_t} \big\langle (\mathbf{H}^{(t)})^{-1}, \mathbf{H}^{(t)} - \mathbf{H}^{(t-1)} + \frac{1}{2}(\sigma_m^{(t-1)})^2 [\mathbf{V}_\perp^{(t-1)}(\mathbf{V}_\perp^{(t-1)})^\top - \mathbf{V}^{(t-1)}(\mathbf{V}^{(t-1)})^\top] \big\rangle \\
\le&\frac{1}{\mu + \eta_T} \sum_{t=1}^{T} \big\langle (\mathbf{H}^{(t)})^{-1}, \mathbf{H}^{(t)} - \mathbf{H}^{(t-1)} + \frac{1}{2}(\sigma_m^{(t-1)})^2 \mathbf{V}^{(t-1)}(\mathbf{V}^{(t-1)})^\top \big\rangle \\
=&\frac{1}{\mu + \eta_T} \sum_{t=1}^{T} \Big[\big\langle (\mathbf{H}^{(t)})^{-1}, \mathbf{H}^{(t)} - \mathbf{H}^{(t-1)} \big\rangle + \frac{1}{2}(\sigma_m^{(t-1)})^2 \mathrm{tr}\big(\mathbf{V}^{(t-1)}(\mathbf{H}^{(t)})^{-1}(\mathbf{V}^{(t-1)})^\top\big) \Big].
\end{aligned}
$$

The above equalities come from the properties of trace operator and (24) and the inequality is due to the fact that η_t is non-increasing.

The term $\sum_{t=1}^{T} \langle (\mathbf{H}^{(t)})^{-1}, \mathbf{H}^{(t)} - \mathbf{H}^{(t-1)} \rangle$ can be bounded as

$$
\begin{aligned}
\sum_{t=1}^{T} \langle (\mathbf{H}^{(t)})^{-1}, \mathbf{H}^{(t)} - \mathbf{H}^{(t-1)} \rangle &\le \sum_{t=1}^{T} \ln \frac{\det(\mathbf{H}^{(t)})}{\det(\mathbf{H}^{(t-1)})} \\
&= \ln \frac{\det(\mathbf{H}^{(T)})}{\det(\mathbf{H}^{(0)})}
\end{aligned}
$$

$$
\begin{aligned}
&= \ln \frac{\prod_{i=1}^{d} \sigma_i(\mathbf{H}^{(T)})}{\alpha_0} \\
&= \sum_{i=1}^{d} \ln \frac{\sigma_i\big((\mathbf{B}^{(T)})^\top \mathbf{B}^{(T)} + \alpha^{(T)}\mathbf{I}_d\big)}{\alpha_0} \\
&= \sum_{i=1}^{m} \ln \frac{\sigma_i\big((\mathbf{B}^{(T)})^\top \mathbf{B}^{(T)}\big) + \alpha^{(T)}}{\alpha_0} + (d-m)\ln \frac{\alpha^{(T)}}{\alpha_0} \\
&\leq m \ln \frac{\sum_{i=1}^{m}[\sigma_i\big((\mathbf{B}^{(T)})^\top \mathbf{B}^{(T)}\big) + \alpha^{(T)}]}{m\alpha_0} + (d-m)\ln \frac{\alpha^{(T)}}{\alpha_0} \\
&= m \ln \Big(\frac{\mathrm{tr}\big((\mathbf{B}^{(T)})^\top \mathbf{B}^{(T)}\big)}{m\alpha_0} + \frac{\alpha^{(T)}}{\alpha_0} \Big) + (d-m)\ln \frac{\alpha^{(T)}}{\alpha_0}.
\end{aligned}
$$

The first inequality is obtained by the concavity of the log determinant function (Boyd and Vandenberghe, 2004), the second inequality comes from the Jensen's inequality and the other steps are based on the procedure of the algorithm.

The other one $\frac{1}{2}\sum_{t=1}^{T}(\sigma_m^{(t)})^2 \mathrm{tr}\big(\mathbf{V}^{(t)}(\mathbf{H}^{(t)})^{-1}(\mathbf{V}^{(t)})^\top\big)$ can be bounded as

$$
\begin{aligned}
\frac{1}{2}\sum_{t=1}^{T}(\sigma_m^{(t)})^2 \mathrm{tr}\big(\mathbf{V}^{(t)}(\mathbf{H}^{(t)})^{-1}(\mathbf{V}^{(t)})^\top\big) &\leq \frac{1}{2}\sum_{t=1}^{T}\frac{(\sigma_m^{(t)})^2}{\alpha^{(t)}} \mathrm{tr}\big(\mathbf{V}^{(t)}(\mathbf{V}^{(t)})^\top\big) \\
&= \frac{m}{2}\sum_{t=1}^{T}\frac{(\sigma_m^{(t)})^2}{\alpha^{(t)}}. \qquad (25)
\end{aligned}
$$

Hence, we have

$$
R_G \leq \frac{1}{\mu+\eta_T}\left[m\ln\Big(\mathrm{tr}\big((\mathbf{B}^{(T)})^\top \mathbf{B}^{(T)}\big) + \frac{\alpha^{(T)}}{\alpha_0}\Big) + (d-m)\ln\frac{\alpha^{(T)}}{\alpha_0} + \frac{m}{2}\sum_{t=1}^{T}\frac{(\sigma_m^{(t)})^2}{\alpha^{(t)}} \right]. \qquad (26)
$$

Then we bound the term R_D by using equation (24), Assumption 1 and Assumption 2.

$$
\begin{aligned}
R_D =& \sum_{t=1}^{T}(\mathbf{w}^{(t)}-\mathbf{w})^\top \big[\eta_t \mathbf{g}^{(t)}(\mathbf{g}^{(t)})^\top + \frac{1}{2}(\sigma_m^{(t-1)})^2\mathbf{I} - \mathbf{V}^{(t-1)}(\mathbf{V}^{(t-1)})^\top\big](\mathbf{w}^{(t)}-\mathbf{w}) \\
\leq& \sum_{t=1}^{T}\eta_t(\mathbf{w}^{(t)}-\mathbf{w})^\top \mathbf{g}^{(t)}(\mathbf{g}^{(t)})^\top(\mathbf{w}^{(t)}-\mathbf{w}) + \frac{1}{2}\sum_{t=1}^{T}(\sigma_m^{(t-1)})^2(\mathbf{w}^{(t)}-\mathbf{w})^\top(\mathbf{w}^{(t)}-\mathbf{w}) \\
\leq& 4(CL)^2\sum_{t=1}^{T}\eta_t + 2C^2\sum_{t=1}^{T}(\sigma_m^{(t-1)})^2. \qquad (27)
\end{aligned}
$$

Finally, we obtain the result by combining (26) and (27). ■

Additionally, the term $\sum_{t=1}^{T}\frac{(\sigma_m^{(t)})^2}{\alpha^{(t)}}$ in Ω_{RFD} can be bounded by $\mathcal{O}(\ln T)$ if we further assume all $\sigma_m^{(t)}$ are bounded by positive constants. Exactly, suppose that $0 < C_1 \leq \big(\sigma_m^{(t)}\big)^2 \leq C_2$ for any

$t = 0, \dots, T-1$, then we have

$$\begin{aligned}
\sum_{t=1}^{T} \frac{(\sigma_m^{(t)})^2}{\alpha^{(t)}} &= \sum_{t=1}^{T} \frac{(\sigma_m^{(t)})^2}{\alpha_0 + \frac{1}{2}\sum_{i=0}^{t-1}\big(\sigma_m^{(i)}\big)^2} \\
&\leq \sum_{t=1}^{T} \frac{C_2}{\alpha_0 + \frac{1}{2}C_1 t} \\
&\leq \frac{2C_2}{C_1}\sum_{t=1}^{T}\frac{1}{t} \\
&= \mathcal{O}(\ln T).
\end{aligned}$$

The last inequality is due to the property of harmonic series.

Appendix H. The Proof of Theorem 6

Considering the update without positive semidefinite assumption on $\mathbf{H}^{(t)}$

$$\begin{aligned}
\mathbf{u}^{(t+1)} &= \mathbf{w}^{(t)} - \beta_t(\mathbf{H}^{(t)})^{\dagger}\mathbf{g}^{(t)}, \\
\mathbf{w}^{(t+1)} &= \underset{\mathbf{w}\in\mathcal{K}_{t+1}}{\operatorname{argmin}} \|\mathbf{w} - \mathbf{u}^{(t+1)}\|_{\mathbf{H}^{(t)}},
\end{aligned} \tag{28}$$

we have the results as follows.

Lemma 7 (Appendix D of Luo et al. (2016)) *Let* $\widehat{\mathbf{H}}^{(t)} = \sum_{s=1}^{t} \mathbf{g}^{(t)}(\mathbf{g}^{(t)})^\top$ *with* $\operatorname{rank}(\widehat{\mathbf{H}}^{(T)}) = r$ *and* σ^* *be the minimum among the smallest non-zeros singular values of* $\widehat{\mathbf{H}}^{(T)}$*. Then the regret of update (28) satisfies*

$$R_T(\mathbf{w}) \leq 2(CL)^2 \sum_{t=1}^{T} \eta^{(t)} + \frac{1}{2}\sum_{t=1}^{T}(\mathbf{g}^{(t)})^\top(\mathbf{H}^{(t)})^{\dagger}\mathbf{g}^{(t)}. \tag{29}$$

and

$$\sum_{t=1}^{T} \mathbf{g}_t^\top(\widehat{\mathbf{H}}^{(t)})^{\dagger}\mathbf{g}_t \leq m - 1 + \frac{m(m-1)}{2}\ln\Big(1 + \frac{2\sum_{t=1}^{T}\|\mathbf{g}^{(t)}\|_2^2}{m(m-1)\sigma^*}\Big) \tag{30}$$

Then we can derive Theorem 6.

Proof For any $t \leq T'$, RFD-SON with $\alpha^{(0)} = 0$ satisfies $\operatorname{rank}(\widehat{\mathbf{B}}^{(t)}) \leq m - 1$. Hence we have

$$\begin{aligned}
\mathbf{H}^{(t)} &= (\mathbf{B}^{(t)})^\top\mathbf{B}^{(t)} \\
&= \sum_{s=1}^{t}\frac{1}{\eta_t + \mu_t}(\mathbf{g}^{(t)})(\mathbf{g}^{(t)})^\top \\
&\succeq \frac{1}{\eta_1 + \mu'}\sum_{s=1}^{t}(\mathbf{g}^{(t)})(\mathbf{g}^{(t)})^\top \\
&= \frac{1}{\eta_1 + \mu'}\widehat{\mathbf{H}}^{(t)}.
\end{aligned} \tag{31}$$

Then we can bound the regret for the first T' iterations

$$\begin{aligned} R_{1:T'}(\mathbf{w}) &\leq 2(CL)^2 \sum_{t=1}^{T'} \eta^{(t)} + \frac{1}{2}\sum_{t=1}^{T'} (\mathbf{g}^{(t)})^\top (\mathbf{H}^{(t)})^\dagger \mathbf{g}^{(t)} \\ &\leq 2(CL)^2 \sum_{t=1}^{T'} \eta^{(t)} + \frac{1}{2(\eta_1+\mu')}\sum_{t=1}^{T'} (\mathbf{g}^{(t)})^\top (\widehat{\mathbf{H}}^{(t)})^\dagger \mathbf{g}^{(t)} \\ &\leq 2(CL)^2 \sum_{t=1}^{T'} \eta^{(t)} + \frac{m-1}{2(\eta_1+\mu')} + \frac{m(m-1)}{2(\eta_1+\mu')} \ln\Big(1 + \frac{2\sum_{t=1}^{T'} \|\mathbf{g}^{(t)}\|_2^2}{m(m-1)\sigma^*}\Big). \end{aligned}$$

The first inequality is based on the inequality (29). The second inequality comes from (31) that is $\mathbf{H}^{(t)} \succeq \frac{1}{\eta_1+\mu'}\widehat{\mathbf{H}}^{(t)}$. And the last one is due to the result (30). ■

Appendix I. The Proof of Theorem 7

Proof Since $\mathbf{H}^{(t)} \succ \mathbf{0}$ for $t \geq T'$, by similar proof of Theorem 5, we have

$$2R_{T'+1:T}(\mathbf{w}) \leq \alpha_0 \|\mathbf{w}^{(T')}\|^2_{\mathbf{H}^{(T')}} + R'_G + R'_D, \tag{32}$$

where

$$\begin{aligned} R'_G =& \sum_{t=T'+1}^{T} (\mathbf{g}^{(t)})^\top (\mathbf{H}^{(t)})^{-1} \mathbf{g}^{(t)} \\ \leq& \frac{1}{\mu+\eta_T}\left[m \ln\Big(\frac{\mathrm{tr}\big((\mathbf{B}^{(T)})^\top \mathbf{B}^{(T)}\big)}{m\alpha^{(T')}} + \frac{\alpha^{(T)}}{\alpha'_0}\Big) + (d-m)\ln\frac{\alpha^{(T)}}{\alpha'_0} + \frac{m}{2}\sum_{t=T'+1}^{T} \frac{(\sigma_m^{(t)})^2}{\alpha^{(t)}} \right], \end{aligned} \tag{33}$$

$$\begin{aligned} R'_D =& \sum_{t=T'+1}^{T} (\mathbf{w}^{(t)} - \mathbf{w})^\top [\mathbf{H}^{(t)} - \mathbf{H}^{(t-1)} - \mu_t \mathbf{g}^{(t)}(\mathbf{g}^{(t)})^\top](\mathbf{w}^{(t)} - \mathbf{w}) \\ \leq& 4(CL)^2 \sum_{t=T'+1}^{T} \eta_t + 2C^2 \sum_{t=T'+1}^{T} (\sigma_m^{(t-1)})^2. \end{aligned} \tag{34}$$

Combining Theorem 6, (32), (33) and (34), we have the final result of (14). ■

References

Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

Dimitris Achlioptas and Frank McSherry. Fast computation of low-rank matrix approximations. *Journal of the ACM*, 54(2):article 9, 2007.

Dimitris Achlioptas, Zohar S. Karnin, and Edo Liberty. Near-optimal entrywise sampling for data matrices. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In *ACM Symposium on Theory of Computing (STOC)*, 2006.

Nir Ailon and Edo Liberty. Fast dimension reduction using Rademacher series on dual BCH codes. *Discrete & Computational Geometry*, 42(4):615–630, 2009.

Nir Ailon and Edo Liberty. An almost optimal unrestricted fast Johnson-Lindenstrauss transform. *ACM Transactions on Algorithms*, 9(3):article 21, 2013.

Sanjeev Arora, Elad Hazan, and Satyen Kale. A fast random sampling algorithm for sparsifying matrices. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. 2006.

Jimmy Ba, Roger Grosse, and James Martens. Distributed second-order optimization using Kronecker-factored approximations. In *International Conference on Learning Representations (ICLR)*, 2017.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Matthew Brand. Incremental singular value decomposition of uncertain data with missing values. In *European Conference on Computer Vision (ECCV)*, 2002.

Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In *ACM Symposium on Theory of Computing (STOC)*, 2013.

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

Petros Drineas and Michael W. Mahoney. On the Nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

Petros Drineas and Anastasios Zouzias. A note on element-wise matrix sparsification via a matrix-valued Bernstein inequality. *Information Processing Letters*, 111(8):385–389, 2011.

Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157, 2006a.

Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36(1):158–183, 2006b.

Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008.

Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, and David P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13: 3475–3506, 2012.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

Murat A. Erdogdu and Andrea Montanari. Convergence rates of sub-sampled Newton methods. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025–1041, 2004.

Mina Ghashami, Edo Liberty, Jeff M. Phillips, and David P. Woodruff. Frequent directions: Simple and deterministic matrix sketching. *SIAM Journal on Computing*, 45(5):1762–1792, 2016.

Roger Grosse and James Martens. A Kronecker-factored approximate fisher matrix for convolution layers. In *International Conference on Machine Learning (ICML)*, 2016.

Ming Gu and Stanley C. Eisenstat. A stable and fast algorithm for updating the singular value decomposition. *Technical Report YALEU/DCS/RR-966*, 1993.

Isabelle Guyon, Steve R. Gunn, Asa Ben-Hur, and Gideon Dror. Result analysis of the NIPS 2003 feature selection challenge. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

Isabelle Guyon, Constantin F. Aliferis, Gregory F. Cooper, André Elisseeff, Jean-Philippe Pellet, Peter Spirtes, and Alexander R. Statnikov. Design and analysis of the causation and prediction challenge. In *Causation and Prediction Challenge at WCCI*, 2008.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53 (2):217–288, 2011.

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

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Elad Hazan and Sanjeev Arora. *Efficient algorithms for online convex optimization and their applications*. Princeton University, 2006.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis, Vol. 2*. Cambridge University Press, 1991.

Zengfeng Huang. Near optimal frequent directions for sketching dense and sparse matrices. In *International Conference on Machine Learning (ICML)*, 2018.

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *ACM Symposium on Theory of Computing (STOC)*, 1998.

William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

Daniel M. Kane and Jelani Nelson. Sparser Johnson-Lindenstrauss transforms. *Journal of the ACM*, 61(1):article 4, 2014.

Avraham Levey and Michael Lindenbaum. Sequential Karhunen-Loeve basis extraction and its application to images. *IEEE Transactions on Image Processing*, 9(8):1371–1374, 2000.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

Edo Liberty. Simple and deterministic matrix sketching. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2013.

Haipeng Luo, Alekh Agarwal, Nicolò Cesa-Bianchi, and John Langford. Efficient second order online learning by sketching. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Michael W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.

James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *International Conference on Machine Learning (ICML)*, 2015.

Andrew McCallum. Sraa: Simulated/real/aviation/auto usenet data. URL `https://people.cs.umass.edu/~mccallum/data.html`.

Chris Mesterharm and Michael J Pazzani. Active learning using on-line algorithms. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2011.

Jayadev Misra and David Gries. Finding repeated elements. *Science of Computer Programming*, 2 (2):143–152, 1982.

Youssef Mroueh, Etienne Marcheret, and Vaibhava Goel. Co-occurring directions sketching for approximate matrix multiply. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

Jelani Nelson and Huy L. Nguyên. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *Symposium on Foundations of Computer Science (FOCS)*, 2013.

Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273, 1982.

Erkki Oja and Juha Karhunen. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of Mathematical Analysis and Applications*, 106(1): 69–84, 1985.

Dimitris Papailiopoulos, Anastasios Kyrillidis, and Christos Boutsidis. Provable deterministic leverage score sampling. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2014.

Mert Pilanci and Martin J. Wainwright. Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence. *SIAM Journal on Optimization*, 27(1):205–245, 2017.

John C. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods: Support Vector Learning*, pages 185–208. Cambridge, MA: MIT Press, 1999.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Farbod Roosta-Khorasani and Michael W. Mahoney. Sub-sampled Newton methods I: globally convergent algorithms. *CoRR*, abs/1601.04737, 2016a.

Farbod Roosta-Khorasani and Michael W. Mahoney. Sub-sampled Newton methods II: local convergence rates. *CoRR*, abs/1601.04738, 2016b.

David A. Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008.

Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *Symposium on Foundations of Computer Science (FOCS)*, 2006.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

Joel A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning*, 8(1-2):1–230, 2015.

Shusen Wang, Luo Luo, and Zhihua Zhang. SPSD matrix approximation vis column selection: Theories, algorithms, and extensions. *Journal of Machine Learning Research*, 17:(49)1–49, 2016.

David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1-2):1–157, 2014.

Peng Xu, Jiyan Yang, Farbod Roosta-Khorasani, Christopher Ré, and Michael W. Mahoney. Sub-sampled Newton methods with non-uniform sampling. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Haishan Ye, Luo Luo, and Zhihua Zhang. Approximate Newton methods and their local convergence. In *International Conference on Machine Learning (ICML)*, 2017.

Qiaomin Ye, Luo Luo, and Zhihua Zhang. Frequent direction algorithms for approximate matrix multiplication with applications in CCA. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

Zhihua Zhang. The matrix ridge approximation: algorithms and applications. *Machine Learning*, 97 (3):227–258, 2014.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning (ICML)*, 2003.