Sketched Iterative Algorithms for Structured Generalized Linear Models

Qilong Gu and **Arindam Banerjee**
Department of Computer Science & Engineering, University of Minnesota, Twin-Cities
{guxxx396, banerjee}@cs.umn.edu

Abstract

Recent years have seen advances in optimizing large scale statistical estimation problems. In statistical learning settings iterative optimization algorithms have been shown to enjoy geometric convergence. While powerful, such results only hold for the original dataset, and may face computational challenges when the sample size is large. In this paper, we study sketched iterative algorithms, in particular sketched-PGD (projected gradient descent) and sketched-SVRG (stochastic variance reduced gradient) for structured generalized linear model, and illustrate that these methods continue to have geometric convergence to the statistical error under suitable assumptions. Moreover, the sketching dimension is allowed to be even smaller than the ambient dimension, thus can lead to significant speed-ups. The sketched iterative algorithms introduced provide an additional dimension to study the trade-offs between statistical accuracy and time.

1 Introduction

In this paper, we consider algorithms for efficiently learning generalized linear models (GLMs) of the form:

$$y_i|x_i \sim \mathbb{P}(y_i|x_i^T\theta^*), \tag{1}$$

where $x_i \in \mathbb{R}^p$ is a data point, $y_i \in \mathbb{R}$ is the response, and $\theta^* \in \mathbb{R}^p$ is the structured parameter that we want to learn. For example, in linear regression $y_i = x_i^T\theta^* + w_i$, where w_i is the noise; in logistic regression $y_i \in \{-1, +1\}$ and $\mathbb{P}(y_i|x_i^T\theta^*) = \frac{1}{1+e^{-y_i x_i^T\theta^*}}$. θ^* can be sparse, low rank, etc [Tibshirani, 1996; Recht *et al.*, 2010]. Estimators for θ^* can be posed as the following constrained problem [Oymak *et al.*, 2018]

$$\min_{R(\theta)\le\lambda} \mathcal{L}_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} l(x_i^T\theta; y_i), \tag{2}$$

where $l : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ is the loss function, and is a function of $x_i^T\theta$ and y_i, e.g., in linear regression, $l(\hat{y}; y)$ is the square loss $\frac{1}{2}(y-\hat{y})^2$, in logistic regression, $l(\hat{y}; y)$ is the logistic loss $\log(1+e^{-y\hat{y}})$, etc., n is the sample size, and $R(.)$ is a suitable norm inducing structures in θ [Banerjee *et al.*, 2014; Negahban *et al.*, 2012], e.g., sparsity can be induced by L_1 norm or k-support norm [Tibshirani, 1996; Argyriou *et al.*, 2012], low-rank can be induced by nuclear norm [Recht *et al.*, 2010], etc. In this paper, we set $\lambda = R(\theta^*)$.

In modern machine learning, problems like (2) are confronted with extremely large datasets, i.e., sample size n is large. Dealing with such large datasets brings about several computational challenges, including large per iteration complexity for methods like gradient descent and memory management challenges for most methods. In this work, inspired by sketching [Woodruff, 2014], a powerful technique for sample size reduction based on random projection, we propose the sketched version of two standard iterative methods, projected gradient descent (PGD) and stochastic variance reduced gradient (SVRG). Since sketching of the data happens once upfront, the iterative algorithms work with a dataset of much smaller effective size.

In this work, we reduce the computation of GLMs while keeping the structure of parameters. For statistical estimation problem (2), at each iteration of PGD and each stage of SVRG, we need to do matrix vector multiplication, and sketching reduces the per iteration time complexity. We show that PGD and SVRG will reach a solution $\hat{\theta}$ such that $\|\hat{\theta} - \theta^*\|_2 \le \frac{C}{n}\left\|\left[\frac{\partial}{\partial\hat{y}} l(x_i^T\theta^*; y_i)\right]_{i=1}^{n}\right\|_2$, for constant $C > 0$. We have two main contributions. First, we present novel sketched iterative algorithms S-PGD and S-SVRG for general loss function of form (2). Our algorithms combine iterative algorithms and sketching. For arbitrary $\delta > 0$, if we choose sketching dimension $m \ge \frac{c}{\delta^2}$ for constant $c > 0$, then the approximate solution $\hat{\theta}$ of our algorithms satisfies

$$\|\hat{\theta} - \theta^*\|_2 \le \frac{1}{\sqrt{n}}\left(C_1\delta + \frac{C_2}{\sqrt{n}}\right)\left\|\left[\frac{\partial}{\partial\hat{y}} l(x_i^T\theta^*; y_i)\right]_{i=1}^{n}\right\|_2,$$

for positive constants C_1 and C_2. Term $\|(\frac{\partial}{\partial\hat{y}} l(x_i^T\theta^*; y_i))_{i=1}^{n}\|_2$ is related to the estimation error of model (1). Different choices of δ determine the accuracy of our approximation. In the literature, progress have been made in applying sketching to least squares [Woodruff, 2014]. However, sketching methods for general optimization problems are limited [Tang *et al.*, 2017; Pilanci and Wainwright, 2016]. [Pilanci and Wainwright, 2017] requires approximating the Hessian of program objective function using sketching in every iteration, which can be computationally expensive . Our algorithms approximate the gradient, which effectively alleviate the computational burden for large-scale problems. To the best of our knowledge, it is the first sketched first-order algorithm for GLMs.

Our second main contribution is to establish the theoretical convergence rates of the proposed sketched algorithms, and experiments on synthetic date to illustrate the theoretical results. We characterize the convergence rates of both S-PGD and S-SVRG, with suitable modifications to standard SVRG. Our results hold for any norm $R(.)$. We show that for sketched algorithms, sketching dimension m is a key factor that determines the tradeoff between computational time and approximation accuracy. We analyze both S-PGD and S-SVRG for constrained least squares and S-PGD for general GLMs. We show that as long as a statistical condition called restricted strong convexity holds, geometric convergence rate can be established even when $m < p$.

The rest of paper is organized as follows. We first review the relevant background and related works in Section 2. We propose our algorithms in Section 3. In Section 4, we discuss convergence of S-PGD and S-SVRG for constrained least squares. In Section 5, we extend our convergence results to general loss function. In Section 6, we compare our algorithm with SGD. We show our algorithms' experimental results and conclude in Section 7 and 8 respectively.

Notations: Throughout this paper, we denote $l'(\hat{y};y) = \frac{\partial}{\partial \hat{y}} l(\hat{y};y)$ and $l''(\hat{y};y) = \frac{\partial^2}{\partial \hat{y}^2} l(\hat{y};y)$, matrix $X = [x_1, \ldots, x_n]^T$ the stack up of all data points, vector $d_n(\theta) = [l'(x_1^T\theta;y_1) \ \ldots \ l'(x_n^T\theta;y_n)]^T$ the first order derivatives of all loss functions, and vector $d_n^{(2)}(\theta) = [l''(x_1^T\theta;y_1) \ \ldots \ l''(x_n^T\theta;y_n)]^T$ the second order derivatives of all loss functions. For a given positive integer n, denote $[n] = \{1, \ldots, n\}$. We use I_n as the $n \times n$ identity matrix. We denote B_n a unit ball and $\mathbb{S}^{n-1}$ a unit sphere in $\mathbb{R}^n$.

2 Background and Related Work

2.1 Projected Gradient Descent (PGD)

One efficient first order algorithm for problem (2) is the projected gradient descent (PGD) [Bertsekas, 2010]. Each step of PGD is given by

$$\theta_{t+1} = P_{\mathcal{K}}(\theta_t - \eta \nabla \mathcal{L}_n(\theta_t)), \tag{3}$$

where η is the learning rate, $P_{\mathcal{K}}(.)$ is the projection operator such that $P_{\mathcal{K}}(z) = \operatorname{argmin}_{y \in \mathcal{K}} \|y - z\|_2^2$, and $\mathcal{K}$ is the constraint set $\mathcal{K} = \{\theta : R(\theta) \leq \lambda\}$. The computation of PGD includes a gradient update and a projection onto feasible set.

2.2 Stochastic Variance Reduction Gradient (SVRG)

If sample size of problem (2) is large, then a popular modification of PGD is stochastic variance reduction gradient (SVRG) [Johnson and Zhang, 2013; Schmidt *et al.*, 2017; Defazio *et al.*, 2014]. Each outer iteration or stage s of SVRG maintains an estimation of solution to (2) θ^s, and a full gradient at θ^s denoted by $\mu_s = \nabla \mathcal{L}_n(\theta^s)$. An inner iteration $t+1$ is given by the following rule

$$\theta_{t+1}^s = P_{\mathcal{K}}(\theta_t - \eta(\nabla_\theta l(x_i^T\theta_t^s;y_i) - \nabla_\theta l(x_i^T\theta^s;y_i) + \mu_s), \tag{4}$$

where index i is drawn uniformly from $[n]$.

2.3 Sub-Gaussian Sketch (SGS)

Given matrices $A \in \mathbb{R}^{n\times p}$ and $B \in \mathbb{R}^{n\times q}$, the time complexity of matrix product A^TB is $O(pqn)$. Sketching approximates A^TB by first sampling a random matrix $S \in \mathbb{R}^{m\times n}$ with $m < n$, and then computing SA and SB. Computing the production $(SA)^T(SB)$ requires only $O(pqm)$.

We say S is a σ-sub-Gaussian random matrix [Vershynin, 2012] if each row of S is an i.i.d. copy of a random vector x, which satisfies $\sup_{u\in\mathbb{S}^{n-1}} P(|\langle u, x\rangle| \geq t) \leq 2e^{-\frac{nt^2}{2\sigma^2}}$. One example of sub-Gaussian random matrix is the Gaussian random matrix, whose each entry is an i.i.d. sample from standard Gaussian distribution. Sub-Gaussian sketch takes $O(mn(p+q))$ basic operations to compute sketched matrices SA and SB. In this paper, we assume $\mathbb{E}S^TS = I_n$.

2.4 Random Orthogonal Sketch (ROS)

In order to construct sketched matrices faster, we consider random orthogonal sketch [Pilanci and Wainwright, 2015]. Let $S = \sqrt{\frac{n}{m}} P \cdot H \cdot D$, where $H \in R^{n\times n}$ is an orthonormal matrix, $P \in \mathbb{R}^{m\times n}$ samples m coordinates of $\mathbb{R}^n$ uniformly at random, and $D \in \mathbb{R}^{n\times n}$ is a diagonal re-randomization matrix of Rademacher random variables. One commonly used ROS is random Hadamard sketch (RHS) [Ailon and Liberty, 2008]. H is called a Hadamard matrix if $H_{i,j} \in \{-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\}$. For any $x \in \mathbb{R}^n$ the mapping Hx can be computed in $O(n \log n)$. RHS takes $O(n(p+q)\log(m))$ basic operations to compute sketched matrices SA and SB.

2.5 Gaussian Width

In this paper, both optimization and statistical results are described in terms of a geometric measure called Gaussian width [Banerjee *et al.*, 2014; Vershynin, 2015; Chandrasekaran *et al.*, 2012]. For any set $\mathcal{A} \in \mathbb{R}^p$, Gaussian width $w(\mathcal{A})$ measures the complexity of $\mathcal{A}$, and it is defined as

$$w(\mathcal{A}) = \mathbb{E}\sup_{u\in\mathcal{A}} \langle u, g\rangle, \tag{5}$$

where $g \sim \mathcal{N}(0, I_n)$ is a standard Gaussian random vector.

2.6 Related Work

Let the minimizer of problem (2) be θ'. Statistical guarantees and computational methods for (2) has been extensively studied. From statistical perspective, much progress has been made in the regime $p \gg n$. [Chandrasekaran *et al.*, 2012] analyzed exact recovery guarantee of atomic norm minimization linear inverse problem in noiseless setting, under restricted strong convexity (RSC) condition [Raskutti *et al.*, 2010]. In noisy setting, [Negahban *et al.*, 2012] established a bound of estimation error $\|\theta' - \theta^*\|_2$ for decomposable norm regularized problem; [Banerjee *et al.*, 2014] analyzed general norm regularized estimator using Gaussian width [Vershynin, 2015].

From the computational perspective, two key ideas have been explored in recent works. Firstly, randomized optimization algorithm has been investigated to reduce computation and storage of large scale problems. One line of research is random projection technique including importance sampling [Mahoney, 2011], sub-Gaussian sketch [Pilanci and Wainwright, 2015], fast Johnson-Lindenstrauss transforms [Ailon and Liberty, 2008], and dimensionality reduction [Zhang *et al.*, 2013;

Algorithm 1: Sketched Projected Gradient Descent (S-PGD)

```
Inputs: $X \in \mathbb{R}^{n \times p}$.
Initialize: $\theta_0 = 0$.
Generate a random matrix $S \in \mathbb{R}^{m \times n}$
Let $\hat{X} = SX$
for $t = 0, 1, \ldots, T$ do
    $\theta_{t+1} = P_{\mathcal{K}}(\theta_t - \eta \hat{X}^T S \cdot d_n(\theta_t))$
end for
```

Wang *et al.*, 2017] etc.. In [Pilanci and Wainwright, 2017], a second-order method called "Newton-sketch" was proposed, which approximates the Hessian of objective function by sketching. [Tang *et al.*, 2017] extended [Pilanci and Wainwright, 2017] to first-order method for constrained least squares. The other line of research is stochastic optimization algorithms. SGD and SVRG are two popular examples [Johnson and Zhang, 2013; Robbins and Monro, 1951].

The second key idea is to use statistical assumptions to establish fast convergence of PGD. Two kinds of convergence are considered in literature: convergence to statistical optimum θ^* and convergence to optimization optimum θ'. [Agarwal *et al.*, 2012] established convergence of PGD by $\|\theta_t - \theta'\|_2 \leq \rho^t \|\theta_0 - \theta'\|_2 + \|\theta' - \theta^*\|_2$. [Qu *et al.*, 2017] extended [Agarwal *et al.*, 2012] to SVRG. In statistical machine learning, a more interesting problem is $\|\theta_t - \theta^*\|_2$. [Oymak *et al.*, 2018] showed that estimation error converge geometrically to an error bound under RSC by PGD. [Li *et al.*, 2016] showed linear convergence of SVRG to statistical error bound with nonconvex constrain.

3 Sketched Iterative Algorithms

In this section, we propose our sketched projected gradient descent and sketched stochastic variance reduction gradient.

For problem (2), one step of PGD algorithm is

$$\theta_{t+1} = P_{\mathcal{K}}(\theta_t - \eta X^T d_n(\theta_t)) \ . \tag{6}$$

For problems with large sample size, the computation of matrix-vector product $X^T \cdot d_n(\theta)$ is not efficient. Given a sketching matrix $S \in \mathbb{R}^{m \times n}$ with $m < n$, we improve the computation by using sketched matrix SX and vector $Sd_n(\theta)$. The sketched algorithm is given by 1. We can compute and store the sketched data matrix $\hat{X} = SX$ upfront and use $\hat{X}$ in every iteration of optimization algorithm, improving the per iteration complexity. The construction of $\hat{X}$ takes $O(mnp)$ for SGS and $O(mp \log n)$ for RHS. For each iteration, we first construct sketched vector $Sd_n(\theta)$, which requires $O(mn)$ for SGS and $O(m \log n)$ for RHS. Then the time complexity of sketched matrix-vector product $X^T S^T S d_n(\theta)$ is $O(pm)$. If we use RHS, then each iteration takes $O(n \log m + pm)$ that is better than $O(pn)$ for direct computation. We call this algorithm sketched projected gradient descent (S-PGD), and it is described as Algorithm 1. For the special case constrained least squares, it is shown [Pilanci and Wainwright, 2015; Tang *et al.*, 2017] that S-PGD 1 finds the solution to the following approximation problem

$$\min_{\theta \in \mathcal{K}} \hat{\mathcal{L}}_n(\theta) = \frac{1}{2n} \|\hat{y} - \hat{X}\theta\|_2^2, \tag{7}$$

where $\hat{y} = Sy$. One iteration of S-PGD reduces per iteration computation from $O(pn)$ to $O(pm)$.

Algorithm 2: Sketched Stochastic Variance Reduction Gradient (S-SVRG)

```
Inputs: $X \in \mathbb{R}^{n \times p}$.
Initialize: $\theta_0 = 0$.
Generate a random matrix $S \in \mathbb{R}^{m \times n}$
Let $\hat{X} = SX$
for $s = 0, 1, \ldots$ do
    $\mu^s = \frac{1}{n} \hat{X}^T S d_n(\theta^s)$
    for $t = 0, 1, \ldots, T$ do
        Randomly select $i$ from $1, \ldots, m$.
        $\theta_{t+1}^s = P_{\mathcal{K}}(\theta_t^s + \eta(\hat{x}_i s_i^T (d_n(\theta_t) - d_n(\theta^s) + \mu^s))$
    end for
    $\theta^{s+1} = \sum_{t=1}^m \frac{1}{2^{m+1-t}} \theta_t^s$
end for
```

By using ideas similar to S-PGD, we propose sketched stochastic variance reduction gradient (S-SVRG) algorithm 2 . Denote $\bar{\theta}_1^s = \theta_1^s$, and we average all the intermediate results by $\bar{\theta}_t^s = \frac{1}{2}\bar{\theta}_{t-1}^s + \frac{1}{2}\theta_t^s$ for all $1 < t \leq m$. At last we output $\theta^{s+1} = \bar{\theta}_m^s$. For large scale problems, S-SVRG applies an one shot sketching, therefore less memory is required in computation. The S-SVRG is given by Algorithm 2 .

4 Constrained Least Squares

In this section, we show the convergence of S-PGD and S-SVRG for constrained least squares. By feasibility, the solution of problem (2) lies in error cone $\mathcal{C}(\theta^*)$, which is the smallest closed cone contains set $\mathcal{K} - \theta$ denoted as $\mathcal{C}(\theta^*) = \text{cone}(\mathcal{K} - \theta^*)$, In this paper, we use $\mathcal{C}$ short for $\mathcal{C}(\theta^*)$. We present both optimization and statistical results in terms of Gaussian width $w(X\mathcal{C} \cap \mathbb{S}^{n-1})$. We make the following assumptions:

Assumption 1 *(Restricted Eigenvelue (RE) Condition) There is a $\kappa_n(\mathcal{C}) > 0$ such that for all $v \in \mathcal{C}$*

$$\tfrac{1}{n}\|Xv\|_2^2 \geq \kappa_n(\mathcal{C})\|v\|_2^2. \tag{8}$$

Assumption 2 *(Boundedness (BD) Condition) There is a $L_n > 0$ such that for all $\theta_1, \theta_2 \in \mathcal{K}$,*

$$\tfrac{1}{n}\|X(\theta_2 - \theta_1)\|_2^2 \leq L_n \|\theta_2 - \theta_1\|_2^2. \tag{9}$$

For constrained least squares, RE and BD conditions corresponds to the RSC and smoothness condition for general loss function. In literature, the RE and BD conditions have been comprehensively studied. [Negahban *et al.*, 2012; Banerjee *et al.*, 2014] have shown that when X is a 1-sub-Gaussian random matrix, then $\kappa_n(\mathcal{C}) \geq 1 - \frac{w^2(\mathcal{C} \cap \mathbb{S}^{p-1})}{n}$ with high probability. [Vershynin, 2012] has shown that when X is sampled from a general distribution, then $L_n = O(1 + c\frac{n}{p})$. We assume X satisfies both RE and BD conditions.

4.1 Convergence of S-PGD

We give the main theoretical result of S-PGD under our assumptions above. Our results precisely characterize the convergence rate and statistical guarantee of S-PGD for constrained least squares estimator .

Theorem 1 *Let $\mathcal{C} = \mathcal{C}(\theta^*)$ and $S \in \mathbb{R}^{m\times n}$ be a σ-sub Gaussian random matrix. Let $\theta^* \in \mathbb{R}^p$ be an arbitrary vector, and $y = X\theta^* + w$. With $\theta_0 = 0$, we apply sketched PGD update $\theta_{t+1} = P_{\mathcal{K}}(\theta_t + \frac{\eta}{n}X^T S^T S(y - X\theta_t))$. Let us assume the RE and BD conditions hold. Set learning rate $\eta = \frac{cmn}{L_n(\sqrt{m}+\sqrt{n})^2}$, if $m > \frac{w^2(X\mathcal{C}\cap\mathbb{S}^{m-1})}{\delta^2}$, then with probability at least $1 - 7\exp\left(-c_1\frac{m\delta^2}{\sigma^4}\right)$ for some $c_1 > 0$*

$$\|\theta_{t+1} - \theta^*\|_2 \leq \left(1 - \frac{m(1-\delta)\kappa_n(\mathcal{C})}{(\sqrt{m}+\sqrt{n})^2 L_n}\right)^t \|\theta^*\|_2 + \frac{1}{\sqrt{n}(1-\delta)\kappa_n}(\sqrt{L_n}\delta + \frac{1}{\sqrt{n}}\xi(\mathcal{C}))\|w\|_2. \tag{10}$$

where $\xi(\mathcal{C}) = \sup_{u\in\mathcal{C}\cap B_n} u^T X^T \frac{w}{\|w\|_2}$.

Theorem 1 shows the precise geometric convergence rate of S-PGD under RSC condition. For sub-Gaussian noise w, we know from literature [Banerjee *et al.*, 2014] that $\|w\|_2 = \Theta(\sqrt{n})$ with high probability. If we choose $\delta = \frac{w(X\mathcal{C}\cap\mathbb{S}^{n-1})}{\sqrt{m}}$, then the second term on the right hand of (10) goes to $\frac{1}{\sqrt{n}\kappa_n}\xi(\mathcal{C})$ with the growth of m. If we also assume X is a sub-Gaussian random matrix, then statistical error bound $\xi(\mathcal{C}) = O(w(\mathcal{C}\cap B_n))$ and RE condition $\kappa_n(\mathcal{C}) = \Omega(1 - \frac{w(\mathcal{C}\cap\mathbb{S}^{p-1})}{n})$ with high probability[Banerjee *et al.*, 2014]. Therefore, as n increases, $\frac{1}{\sqrt{n}\kappa_n}\xi(\mathcal{C})$ goes down to zero. The result of [Oymak *et al.*, 2018] shows that the convergence rate $\rho(\mathcal{C})$ of PGD is $1 - \frac{\kappa_n(\mathcal{C})}{L_n}$, thus S-PGD have slower convergence rate $\rho(\mathcal{C})$ due to the effect of sketching. But each update of S-PGD can be computed more efficiently.

4.2 Convergence of S-SVRG

We characterize convergence rate for constrained least squares using S-SVRG. Our conclusions precisely describe how sketching impacts the convergence rate of S-SVRG and the statistical error bound of our estimator. The convergence of S-SVRG is presented in the following theorem:

Theorem 2 *Let $\mathcal{C} = \mathcal{C}(\theta^*)$, $S \in \mathbb{R}^{m\times n}$ be a σ-sub Gaussian random matrix. Let $\theta^* \in \mathbb{R}^p$ be an arbitrary vector, and $y = X\theta^* + w$. With $\theta_0 = 0$, we apply sketched SVRG update $\theta^s_{t+1} = P_{\mathcal{K}}(\theta^s_t + \eta(\hat{x}_i\hat{x}_i^T(\theta^s - \theta^s_t) - \hat{\mu}^s))$. Denote the i-th row of S as s_i, $\hat{x}_i$ is given by $\hat{x}_i = \frac{1}{\sqrt{n}}X^T s_i$. Let us assume the RE and BD conditions, and $\frac{L_n}{(1-\delta)\kappa_n(\mathcal{C})} \leq \frac{3}{2}$ hold. Set learning rate $\eta = \frac{1}{2L_n}$, if $m > \frac{w^2(X\mathcal{C}\cap\mathbb{S}^{m-1})}{\delta^2}$, then starting from $\theta^0 = 0$, with probability at least $1 - 7\exp\left(-c_1\frac{m\delta^2}{\sigma^4}\right)$,*

$$\mathbb{E}\|\theta^{s+1} - \theta^*\|_2^2 \leq \left(\frac{7}{8}\right)^s \frac{(\sqrt{m}+\sqrt{n})^2 L_n}{(1-\hat{\delta})\kappa_n(\mathcal{C})}\|\theta^*\|_2^2 + \frac{C}{(1-\delta)\kappa_n(\mathcal{C})L_n n}(\sqrt{L_n}\delta + \frac{1}{\sqrt{n}}\xi(\mathcal{C}))^2\|w\|_2^2, \tag{11}$$

$\xi(\mathcal{C}) = \sup_{u\in\mathcal{C}\cap B_n} u^T X^T \frac{w}{\|w\|_2}$ for constants $c_1, C > 0$.

Theorem 2 indicates the linear convergence of S-SVRG. Similar to PGD, sketching brings approximation error $\sqrt{L_n}\delta$. To get better approximation solution in other words smaller δ, we should increase sketching dimension m such that $m > \frac{w^2(X\mathcal{C}\cap\mathbb{S}^{m-1})}{\delta^2}$. We also need to choose $\delta \leq 1 - \frac{2L_n}{3\kappa_n(\mathcal{C})}$ such that bound (11) is valid.

Example (Lasso): If the structured parameter θ^* is s-sparse, we can use l_1 norm ball as constraint. l_1 constrained least squares is also known as Lasso [Tibshirani, 1996].

Let $S \in \mathbb{R}^{m\times n}$ be a σ-sub Gaussian random matrix. Let $\theta^* \in \mathbb{R}^p$ be an s-sparse vector, and $y = X\theta^* + w$. With $\theta_0 = 0$, we apply sketched PGD update. Let us assume the RSC and BD conditions hold. Take $\eta = \frac{cmn}{L_n(\sqrt{m}+\sqrt{n})^2}$, if $m > c\cdot\frac{s\log p}{\delta^2}$ for some $c > 0$ [Banerjee *et al.*, 2014; Pilanci and Wainwright, 2015], then starting from $\theta^0 = 0$, with high probability

$$\|\theta_{t+1} - \theta^*\|_2 \leq \left(1 - \frac{m(1-\delta)\kappa_n(\mathcal{C})}{(\sqrt{m}+\sqrt{n})^2 L_n}\right)^t \|\theta^*\|_2 + \frac{1}{\sqrt{n}(1-\delta)\kappa_n(\mathcal{C})}(\sqrt{L_n}\delta + \frac{1}{\sqrt{n}}\xi(\mathcal{C}))\|w\|_2. \tag{12}$$

Example (Low-Rank Matrix Regression): In low-rank matrix regression, the true parameter $\Theta^* \in \mathbb{R}^{d\times p}$ is a rank r matrix with $r \ll \min\{d, p\}$. n observations can be accessed by the noisy linear model $y_i = \langle X_i, \Theta^*\rangle + w_i$, where $\langle .,.\rangle$ denotes the entry wise inner product of two matrices, and X_i is a data point. One commonly used constraint is nuclear norm ball [Recht *et al.*, 2010]. Denote $\|.\|_*$ the nuclear norm, we set $R(\theta) = \|\theta\|_*$. Let us assume the RE and BD conditions hold. Take $\eta = \frac{cmn}{L_n(\sqrt{m}+\sqrt{n})^2}$, if $m > c\cdot\frac{r(d+p)}{\delta^2}$ for some $c > 0$ [Chandrasekaran *et al.*, 2012], then starting from $\Theta^0 = 0$, with high probability

$$\|\Theta_{t+1} - \Theta^*\|_2 \leq \left(1 - \frac{m(1-\delta)\kappa_n(\mathcal{C})}{(\sqrt{m}+\sqrt{n})^2 L_n}\right)^t \|\Theta^*\|_2 + \frac{1}{\sqrt{n}(1-\delta)\kappa_n(\mathcal{C})}(\sqrt{L_n}\delta + \frac{1}{\sqrt{n}}\xi(\mathcal{C}))\|w\|_2. \tag{13}$$

5 General Loss Functions

In previous sections, we have discussed sketched iterative algorithms for constrained least squares. Unlike sketched least squares, Algorithm 1 does not solve a specific optimization problem. Instead, we find that if Algorithm 1 converges, the convergent point will be a solution to a system of nonlinear equations. The conclusion follows Lemma 3.

Lemma 3 *Assume there is an $\eta > 0$ and a $\gamma < 1$ such that*

$$\|\theta_2 - \theta_1 - \eta X^T S^T S(d_n(\theta_2) - d_n(\theta_1))\|_2 \leq \gamma\|\theta_2 - \theta_1\|_2,$$

for all $\theta_1, \theta_2 \in \mathbb{R}^p$, then starting from arbitrary point, the iteration $\theta_{t+1} = \theta_t - \eta X^T S^T S\cdot d_n(\theta_t)$, has a limitation $\hat{\theta}$ which satisfies $X^T S^T S d_n(\hat{\theta}) = 0$.

From Lemma 3, S-PGD approximates first order optimality condition $X^T d_n(\theta) = 0$, which is the same as least squares.

If sketching dimension m is large enough, then $S^T S$ will be approximately identity, and we would expect $\hat{\theta}$ and (2)'s optimum to be very close.

Our results requires the following assumptions:

Assumption 3 *(Strong Convexity and Smoothness of Loss Function) All $l(.;y_i)$ are strongly convex and smooth. That is there is an $\alpha > 0$ and a $\beta > 0$ such that $\alpha \leq l''(x_i^T\theta; y_i) \leq \beta$, for any $i \in [n], \theta \in \mathcal{K}$.*

Assumption 4 *There is a $0 < \delta_1 < 1$ such that for any $u \in X\mathcal{C}$, and diagonal matrix $D = diag(d)$ with $d \in [\frac{\alpha}{\beta}, 1]^n$*

$$|u^T D(S^T S - I)Du| \leq \frac{\alpha}{3\beta}\delta_1. \tag{14}$$

Assumption 5 *There is a $\delta_2 > 0$ such that for any $u \in X\mathcal{C} \cap B_n$, $v \in span(X) \cap B_n$, and diagonal matrix $D = diag(d)$ with $d \in [\frac{\alpha}{\beta}, 1]^n$, where $span(X)$ is the subspace spanned by the columns of X*

$$|u^T D(S^T S - I)v| \leq \delta_2. \tag{15}$$

Assumptions 4 and 5 implies that matrix $S^T S$ is close to identity in some scale cones. From literature [Pilanci and Wainwright, 2017; Pilanci and Wainwright, 2015; Woodruff, 2014], sketch dimension m determines the size of δ_1 and δ_2 in assumptions 5, 4. Assumption 4 also indicates that m should be in proportional to loss function condition number $\frac{\beta}{\alpha}$.

We then characterize the convergence of S-PGD.

Theorem 4 *Let θ^* be an arbitrary vector, and y is generated according to* (1). *With $\theta_0 = 0$, we apply S-PGD. Let us assume the RE and BD conditions hold for X. Set learning rate $\eta = \frac{\alpha(1-\delta_2)\kappa_n(\mathcal{C})}{\beta^2(1+\delta_2)^2 L_n^2}$, if assumptions 3, 5 , and 4 hold, we have at step $t+1$*

$$\|\theta_{t+1} - \theta^*\|_2 \leq \rho(\mathcal{C}; S)\|\theta_t - \theta^*\|_2 + \eta\xi(\mathcal{C}; S)\|d_n(\theta^*)\|_2,$$

where $\rho(\mathcal{C}; S) = \left(1 - \frac{\alpha^2(1-\delta_1)^2\kappa_n^2(\mathcal{C})}{\beta^2(1+\delta_2)^2 L_n^2}\right)^{\frac{1}{2}}$, and $\xi(\mathcal{C}; S) = \sup_{u\in\mathcal{C}\cap B} u^T X^T S^T S \frac{d_n(\theta^*)}{\|d_n(\theta^*)\|_2}$.

In Theorem 4 we use the same RE and BD conditions as least squares (Theorem 1). Note that in general case S-PGD convergence rate $\rho(\mathcal{C}; S) = \sqrt{1 - \frac{\kappa_n^2(\mathcal{C})}{L_n^2}}$ is slightly worse than $\rho(\mathcal{C}) = 1 - \frac{\kappa_n(\mathcal{C})}{L_n}$ of least squares (Theorem 1) .

Example (Structured Logistic Regression): In structured logistic regression, its loss function is logistic loss

$$l(x_i^T\theta; y_i) = \log(1 + e^{-y_i x_i^T\theta}), \tag{16}$$

where label $y_i \in \{+1, -1\}$. If the structured parameter θ^* can be captured by norm $R(.)$, then the optimization problem is posed as following

$$\min_\theta \frac{1}{n}\sum_{i=1}^n \log(1 + e^{-y_i x_i^T\theta}) \quad s.t. \quad R(\theta) \leq R(\theta^*). \tag{17}$$

The convexity and smoothness constant in Assumption 3 can be bounded by the following conclusion

Lemma 5 *When $l(x_i^T\theta; y_i)$ is the logistic loss* (16)*, we have $\alpha \leq l''(x_i^T\theta; y_i) \leq \frac{1}{4}$, where $\alpha = ((1 + e^{-MR(\theta^*)})(1 + e^{MR(\theta^*)}))^{-1}$ for all i and $\theta \in \mathcal{K}$, where $M = \max_i R^*(x_i)$, and $R^*(.)$ is the dual norm of $R(.)$.*

Let $S \in \mathbb{R}^{m\times n}$ be a sketching matrix, the main step of S-PGD for (17) is $\theta_{t+1} = P_{R(\theta)\leq R(\theta^*)}(\theta_t - \frac{\eta}{n}(SX)^T S \cdot d_n(\theta_t))$, where $d_{n,i}(\theta_t) = \frac{-y_i}{1+e^{y_i x_i^T\theta_t}}$. Let us assume there exists a RSC constant $\kappa_n(\mathcal{C}) > 0$ and smoothness constant $L_n > 0$, where error cone $\mathcal{C} = \text{cone}\{\Delta : R(\theta^* + \Delta) \leq R(\theta^*)\}$. Take $\eta = \frac{\alpha(1-\delta_1)\kappa_n(\mathcal{C})}{\beta^2(1+\delta_2)^2 L_n^2}$, then starting from $\theta^0 = 0$,

$$\begin{aligned}&\|\theta_{t+1} - \theta^*\|_2 \leq \rho^t(\mathcal{C}; S)\|\theta^*\|_2 \\ &+ \frac{1}{\sqrt{n}(1 - \rho(\mathcal{C}; S))}(\sqrt{L_n}\delta + \frac{1}{\sqrt{n}}\xi(\mathcal{C}))\|d_n(\theta^*)\|_2,\end{aligned} \tag{18}$$

where $\rho(\mathcal{C}; S) = \left(1 - \frac{\alpha^2(1-\delta_1)^2\kappa_n^2(\mathcal{C})}{\beta^2(1+\delta_2)^2 L_n^2}\right)^{\frac{1}{2}}$.

6 Comparison with SGD

In this section, we compare S-PGD with SGD. SGD can be treated a special case of S-PGD. We introduce leverage score and discuss how different types of sketching affects the leverage score. Finally, we compare the performance of SGD and Hadamard sketching using leverage score.

Let $S = \sqrt{\frac{n}{m}}P \cdot I_n$, where $I_n \in \mathbb{R}^n$ is the identity matrix, then one iteration of SGD corresponds to S-PGD using the S defined above. In each iteration, SGD sample a new sketching matrix S in every iteration, while S-PGD only sample one S.

Let us consider ROS without re-randomization: $S = \sqrt{\frac{n}{m}}P \cdot H$, we show that the performance of different S can be measured by leverage score. For S-PGD (1), the i-th leverage score [Mahoney, 2011] is defined as $p_i(\theta) = \frac{\|h_i^T X\|_2^2 (h_i^T d(\theta))^2}{\sum_{j=1}^n \|h_j^T X\|_2^2 (h_j^T d(\theta))^2}$, where h_i is the i-th row of H. Let $\mu(\theta) = n \cdot \max_i p_i(\theta)$, we have the following approximation bound from [Mahoney, 2011; Drineas *et al.*, 2006]

Theorem 6 *Let $\delta \in (0, 1)$, with probability at least $1 - \delta$, the difference between gradient and sketching gradient can be bounded by*

$$\frac{\|X^T d(\theta) - X^T S^T S d(\theta)\|_2^2}{\|X\|_F^2 \|d(\theta)\|_2^2} = O\left(\frac{\mu^2(\theta)\log(1/\delta)}{m}\right). \tag{19}$$

From Theorem 6, the performance of SGD depends on $\mu(\theta)$. When $d(\theta)$ is close to sparse, $\mu(\theta)$ is large and sketching update $X^T S^T S d(\theta)$ will not work. We set H to be the Hadamard matrix and re-randomize S using D, then $X^T S^T S d(\theta)$ will provide a desired stochastic gradient,

7 Experimental Results

In this section, we show the experimental results of our algorithms on synthetic dataset, and how the choice of m affects computational efficiency and statistical guarantee.

In our first set of experiments, we show how m affects the performance of our algorithm using least squares. We draw design matrix $X \in \mathbb{R}^{n\times p}$ randomly from Gaussian distribution. We choose parameter θ^* to be an s-sparse vector, non-zero entries of θ^* are drawn from standard Gaussian distribution. Response y is given by $y = X\theta^* + \sigma \cdot w$, where $\sigma > 0$ is a constant and w is drawn from standard Gaussian $\mathcal{N}(0, 1)$. We sample sketching matrix S from $\mathcal{N}(0, \frac{1}{\sqrt{m}})$ elementwisely.

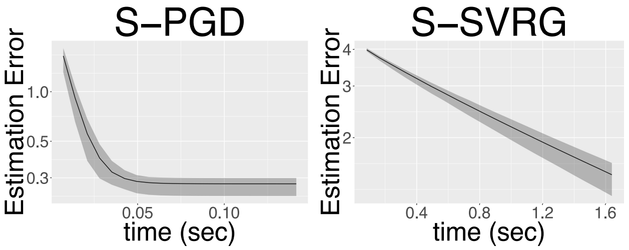

Figure 1: Plot of running time versus estimation error $\|\theta_t - \theta^*\|_2$ for PGD (left) and SVRG (right). We set sample size n to be 2^{17} and dimension p to be 2000. We set sketching dimension m to be 1000. Estimation error is presented in log scale. We can see that both algorithms converge linearly.

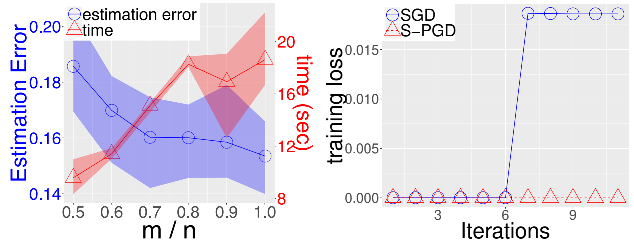

Figure 2: (left) Dimension reduction ratio $\frac{m}{n}$ versus running time (left, blue) and statistical error $\|\theta_t - \theta^*\|_2$ of least squares (right, red) using S-PGD. As we increase the size of m, S-PGD uses more time to converge but the output becomes more accurate. (right) Comparison of S-PGD and SGD on our designed synthetic dataset start from special initial point. S-PGD converges while SGD increases the training loss.

In the first experiment, we choose $n = 131,072(2^{17})$, $p = 2000$, $m = 1000$, and $s = 10$. We plot mean running time for each iteration versus mean estimation error $\|\theta_t - \theta^*\|_2$ and error bar of estimation error in Figure 1. We can see that both algorithms converge linearly. In the second experiment we choose $n = 20000$, $p = 5000$, $s = 10$, and $m \in \{20000, 18000, 16000, 14000, 12000, 10000\}$. For each m we run S-PGD 10 times. We stop our algorithm when relative error $\frac{\|\theta_{t+1} - \theta_t\|_2}{\|\theta_t\|_2}$ is smaller than 0.01. We plot ratio $\frac{m}{n}$ versus estimation error $\|\hat{\theta} - \theta^*\|_2$ and running time for each iteration, estimation error in Figure 2 left. We can see that as m increases, S-PGD converges slower but the output becomes more accurate.

In our second set of experiments, we compare RHS based S-PGD with PGD for binary logistic regression. Each data point (y_i, x_i) is generated by $x_i \sim \mathcal{N}(0, I_p)$, $\mathbb{P}(y_i|x_i^T\theta^*) = \frac{1}{1+e^{-y_i x_i^T \theta^*}}$. Let $p = 1000$, $n = 1,048,576(2^{20})$, and $m = 2000$. We run 900 iterations for both algorithms. We collect the running time and estimation error every 10 iterations. The experimental results are shown in Figure 3. We can see that PGD takes less iterations to reach a smaller error, but in about the first 600 seconds S-PGD outperforms PGD. In summary, PGD has faster convergence rate but slower per iteration computation than S-PGD. S-PGD do more iterations in less time than PGD.

In our third set of experiments, we compare RHS based S-PGD with SGD for binary logistic regression. We choose $n = 131,072(2^{17})$, $p = 1000$, and $m = 1000$. Firstly, we sample uniformly from the unit sphere $\mathbb{S}_{p-1} \subset \mathbb{R}^p$ a vector θ_0. Denote $\mathbb{S}_{p-1}^+$ a sphere gap such that for all $u \in \mathbb{S}_{p-1}^+$ the angle between θ_0 and u is in $\left[0, \frac{\pi}{3}\right]$. Let $y \in \{1, -1\}^n$ be the label vector and $X \in \mathbb{R}^{n \times p}$ be the data matrix. We set $y_i = 1$ and sample x_i uniformly from $\mathbb{S}_{p-1}^+$ if $i = 2, \ldots, 65,536(2^{16})$; We set $y_i = -1$ if $i = 65,538, \ldots, 131,072$ and sample x_i uniformly from $-\mathbb{S}_{p-1}^+ = \{v : -v \in \mathbb{S}_{p-1}^+\}$. Then we set $y_1 = 1$, $y_{65537} = -1$ and randomly sample x_1 and x_{65537} such that $x_i = 1000 \cdot g_i, i = 1, 65,537$ where $g_i \in \mathbb{R}^p$ is a random vector that satisfies $g_i^T \theta_0 = 0$. Finally, we constructed a dataset with two outliers. We set sketching dimension $m = 1$ and initialize both algorithms by w_0. We compare the performance of S-PGD and SGD in terms of $\mathcal{L}_n(\theta)$ and iterations. We run both S-PGD and SGD 100 times, and plot the average of training loss versus number of iterations.

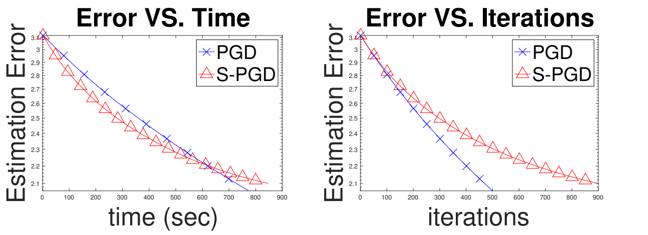

Figure 3: Convergence comparison of PGD and S-PGD. For both algorithms we plot running time versus estimation error $\|\theta_t - \theta^*\|_2$ (left) and number of iterations versus estimation error (right). We set sample size n to be 2^{20} and dimension p to be 1000. We set sketching dimension m to be 2000. We can see from the left that in about the first 600 seconds S-PGD achieves lower estimation error. By comparing the result of both left and right we can see that S-PGD has faster per iteration computation.

From Figure 2 right, the loss function trained by S-PGD keeps going down while SGD stops, which implies that SGD failed to find an effective sample in fixed iterations. Vector $d(\theta_0)$ is close to sparse since only two samples are misclassified, and $\mu(\theta_0)$ is large because the misclassified samples have greater $\|x\|_2$. RHS based S-PGD finds descent directions and move towards the real optimum.

8 Conclusion

In this paper we discussed sketching based iterative algorithms for generalized linear models. We show that under proper assumptions, these algorithms converge linearly to an error bound. The rate of convergence and error bound is determined by size of random projection and property of design matrix. In future works we want to explore if there exists similar results for nonlinear problems.

Acknowledgments

The research was also supported by NSF grants IIS-1563950, IIS-1447566, IIS-1447574, IIS-1422557, CCF-1451986, CNS- 1314560, IIS-0953274, IIS-1029711, NASA grant NNX12AQ39A, and gifts from Adobe, IBM, and Yahoo.

References

[Agarwal *et al.*, 2012] Alekh Agarwal, Sahand Negahban, and Martin J. Wainwright. Fast global convergence of gradient methods for high-dimensional statistical recovery. *The Annals of Statistics*, 40(5):2452–2482, 10 2012.

[Ailon and Liberty, 2008] Nir Ailon and Edo Liberty. Fast dimension reduction using Rademacher series on dual BCH codes. *Discrete & Computational Geometry*, 42(4):615, Sep 2008.

[Argyriou *et al.*, 2012] Andreas Argyriou, Rina Foygel, and Nathan Srebro. Sparse Prediction with the k-Support Norm. In *NIPS*, pages 1457–1465, 2012.

[Banerjee *et al.*, 2014] Arindam Banerjee, Sheng Chen, Farideh Fazayeli, and Vidyashankar Sivakumar. Estimation with Norm Regularization. In *NIPS*, pages 1556–1564, 2014.

[Bertsekas, 2010] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2010.

[Chandrasekaran *et al.*, 2012] Venkat Chandrasekaran, Benjamin Recht, Pablo A. Parrilo, and Alan S. Willsky. The Convex Geometry of Linear Inverse Problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

[Defazio *et al.*, 2014] Aaron Defazio, Francis Bach, and Simon Lacoste-julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*, pages 1646–1654, 2014.

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

[Johnson and Zhang, 2013] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, pages 315–323, 2013.

[Li *et al.*, 2016] Xingguo Li, Tuo Zhao, Raman Arora, Han Liu, and Jarvis Haupt. Stochastic variance reduced optimization for nonconvex sparse learning. In *ICML*, pages 917–925, 2016.

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

[Negahban *et al.*, 2012] Sahand N. Negahban, Pradeep Ravikumar, Martin J. Wainwright, and Bin Yu. A Unified Framework for High-Dimensional Analysis of M-Estimators with Decomposable Regularizers. *Statistical Science*, 27(4):538–557, 2012.

[Oymak *et al.*, 2018] Samet Oymak, Benjamin Recht, and Mahdi Soltanolkotabi. Sharp time–data tradeoffs for linear inverse problems. *IEEE Transactions on Information Theory*, pages 4129–4158, 2018.

[Pilanci and Wainwright, 2015] Mert Pilanci and Martin J. Wainwright. Randomized sketches of convex programs with sharp guarantees. *IEEE Transactions on Information Theory*, pages 5096–5115, 2015.

[Pilanci and Wainwright, 2016] Mert Pilanci and Martin J. Wainwright. Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares. *Journal of Machine Learning Research (JMLR)*, 17(1):1842–1879, January 2016.

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

[Qu *et al.*, 2017] Chao Qu, Yan Li, and Huan Xu. Linear Convergence of SVRG in Statistical Estimation. *arXiv:1611.01957*, 2017.

[Raskutti *et al.*, 2010] Garvesh Raskutti, Martin J. Wainwright, and Bin Yu. Restricted Eigenvalue Properties for Correlated Gaussian Designs. *Journal of Machine Learning Research (JMLR)*, 11:2241–2259, 2010.

[Recht *et al.*, 2010] Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[Robbins and Monro, 1951] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 09 1951.

[Schmidt *et al.*, 2017] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1):83–112, Mar 2017.

[Tang *et al.*, 2017] Junqi Tang, Mohammad Golbabaee, and Mike E. Davies. Gradient projection iterative sketch for large-scale constrained least-squares. In *ICML*, pages 3377–3386, 2017.

[Tibshirani, 1996] Robert Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society*, 58(1):267–288, 1996.

[Vershynin, 2012] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In *Compressed Sensing*, pages 210–268. Cambridge University Press, Cambridge, nov 2012.

[Vershynin, 2015] Roman Vershynin. *Estimation in High Dimensions: A Geometric Perspective*, pages 3–66. Springer International Publishing, Cham, 2015.

[Wang *et al.*, 2017] Jialei Wang, Jason Lee, Mehrdad Mahdavi, Mladen Kolar, and Nati Srebro. Sketching Meets Random Projection in the Dual: A Provable Recovery Algorithm for Big and High-dimensional Data. In *AISTATS*, pages 1150–1158, 2017.

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

[Zhang *et al.*, 2013] Lijun Zhang, Mehrdad Mahdavi, Rong Jin, Tianbao Yang, and Shenghuo Zhu. Recovering the Optimal Solution by Dual Random Projection. In *COLT*, pages 135–157. 2013.