# Solving Ridge Regression using Sketched Preconditioned SVRG

**Alon Gonen** ALONGNN@CS.HUJI.AC.IL
The Hebrew University

**Francesco Orabona** FRANCESCO@ORABONA.COM
Yahoo Research, 229 West 43rd Street, 10036 New York, NY, USA

**Shai Shalev-Shwartz** SHAIS@CS.HUJI.AC.IL
The Hebrew University

## Abstract

We develop a novel preconditioning method for ridge regression, based on recent linear sketching methods. By equipping Stochastic Variance Reduced Gradient (SVRG) with this preconditioning process, we obtain a significant speed-up relative to fast stochastic methods such as SVRG, SDCA and SAG.

## 1. Introduction

Consider the *ridge regression* problem:

$$\min_{w\in\mathbb{R}^d} \left\{ L(w) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}(w^\top x_i - y_i)^2 + \frac{\lambda}{2}\|w\|^2 \right\}, \quad (1)$$

where $\lambda > 0$ is a regularization parameter, $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$ for $i = 1, \cdots, n$ the training data. We focus on the large scale regime, where both $n$ and $d$ are large. In this setting, stochastic iterative methods such as SDCA (Shalev-Shwartz & Zhang, 2013), SVRG (Johnson & Zhang, 2013), and SAG (Roux et al., 2012) have become a standard choice for minimizing the objective $L$. Specifically, the overall complexity of a recent improved variant of SVRG due to Xiao & Zhang (2014) depends on the average condition number, which is defined as follows. Denote the empirical correlation matrix and its eigenvalue decomposition by

$$C := \frac{1}{n}\sum_{i=1}^{n} x_i x_i^\top = \sum_{i=1}^{d} \lambda_i u_i u_i^\top .$$

The average condition number of $C + \lambda I$ is defined as the ratio between the trace of the Hessian of $L$ and its minimal eigenvalue:

$$\hat\kappa := \hat\kappa(C+\lambda I) = \frac{\mathrm{tr}(C+\lambda I)}{\lambda_d(C+\lambda I)} = \sum_{i=1}^{d} \frac{\lambda_i + \lambda}{\lambda_d + \lambda} .$$

The mentioned variant of SVRG finds an $\epsilon$-approximate minimizer of $L$ in time $\tilde O((\hat\kappa + n)d\log(1/\epsilon))$. Namely, the output of the algorithm, denoted $\hat w$, satisfies $\mathbb{E}[L(\hat w)] - L(w^\star) \le \epsilon$, where the expectation is over the randomness of the algorithm. For an accelerated version of the algorithm, we can replace $\hat\kappa$ by $\sqrt{n\hat\kappa}$ (Shalev-Shwartz & Zhang, 2014; Lin et al., 2015).

The regularization parameter, $\lambda$, increases the smallest eigenvalue of $C+\lambda I$ to be at least $\lambda$, thus improves the condition number and makes the optimization problem easier. However, to control the under/over fitting tradeoff, $\lambda$ has to decrease as $n$ increases (Shalev-Shwartz & Ben-David, 2014). Moreover, in many machine learning applications $\lambda_d$ approaches zero and it is usually smaller than the value of $\lambda$. Overall, this yields a large condition number in most of the interesting cases.

A well-known approach for reducing the average condition number is *preconditioning*. Concretely, for a (symmetric) positive definite (pd) matrix $P \in \mathbb{R}^{d\times d}$, we define the preconditioned optimization problem as

$$\min_{\tilde w\in\mathbb{R}^d} \tilde L(\tilde w) := L(P^{-1/2}\tilde w) . \quad (2)$$

Note that $\tilde w$ is an $\epsilon$-approximate minimizer of $\tilde L$ if and only if $w = P^{-1/2}\tilde w$ forms an $\epsilon$-approximate minimizer of $L$. Hence, we can minimize (2) rather than (1). As we shall see, the structure of the objective allows us to apply the preconditioning directly to the data (as a preprocessing step) and consequently rewrite the preconditioned objective as a ridge regression problem with respect to the preconditioned data (see Section 5.1). For a suitable choice of a matrix $P$, the average condition number is significantly reduced. Precisely, as will be apparent from the analysis, the

*Proceedings of the 33rd International Conference on Machine Learning*, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

pd matrix that minimizes the average condition number is $P = C + \lambda I$, and the corresponding average condition number is $d$. However, we note that such preconditioning process would require both the computation of $P^{-1/2}$ and the computation of $P^{-1/2}x_i$ for each $i \in [n]$. By first order conditions, computing $(C + \lambda I)^{-1/2}$ is equivalent to solving the original problem in (1), rendering this "optimal" preconditioner useless.

Yet, the optimal preconditioner might not needed in many cases. In fact, a common empirical observation (see Section 6) is that (high-dimensional) machine learning problems tend to have few dominant features, while the other coordinates are strongly correlated with the stronger features. As a result, the spectrum of the correlation matrix decays very fast. Hence, it is natural to expect to gain a lot from devising preconditioning methods that focus on the stronger directions of the data.

Our contributions are as follows. We develop a relatively cheap preconditioning method that, coupled with SVRG, assures to speed-up the convergence in practical applications while having a computational cost comparable to SVRG alone. In order to approximately extract the stronger directions while incurring a low computational cost, we rely on a variant of the Block Lanczos method due to Musco & Musco (2015) in order to compute an approximated truncated SVD (Singular Value Decomposition) of the correlation matrix $C$. Finally, by equipping SVRG with this preconditioner, we obtain our main result.

## 2. Main Result

**Theorem 1.** *Let $k \in [d]$ be a given parameter and assume that the regularization parameter, $\lambda$, is larger than $\lambda_d$. Our preconditioning process runs in time $O(ndk\log(n))$. By equipping the SVRG of Xiao & Zhang (2014) with this preconditioner, we find an $\epsilon$-approximate minimizer for* (1) *(with probability at least $9/10$) in additional runtime of $O((\tilde{\kappa} + n + d)d\log(1/\epsilon))$, where $\tilde{\kappa} = \frac{k\lambda_k + \sum_{i>k}\lambda_i}{\lambda}$ or $\tilde{\kappa} = \left(\frac{n(k\lambda_k + \sum_{i>k}\lambda_i)}{\lambda}\right)^{1/2}$ if we use accelerated SVRG.*

When the runtimes of both the (accelerated) SVRG and our preconditioned (accelerated) SVRG are controlled by the average condition number (and both runtimes dominate $ndk$), then ignoring logarithmic dependencies, we obtain a speed-up of order

$$\text{ratio} = \frac{\sum_{i=1}^{d}\lambda_i}{k\,\lambda_k + \sum_{i>k}\lambda_i} = \frac{\sum_{i=1}^{k}\lambda_i + \sum_{i>k}\lambda_i}{k\,\lambda_k \quad + \sum_{i>k}\lambda_i} . \quad (3)$$

(or $\sqrt{\sum_{i=1}^{d}\lambda_i/(\lambda_k k + \sum_{i>k}\lambda_i)}$ if acceleration is used) over SVRG. If the spectrum decays fast then $k\,\lambda_k \ll \sum_{i=1}^{k}\lambda_i$ and $\sum_{i>k}\lambda_i \ll k\,\lambda_k$. In this case, the ratio will be large. Indeed, as we show in the experimental section, this ratio is often *huge* for relatively small $k$.

### 2.1. Main challenges and perspective

While the idea of developing a preconditioner that focuses on the stronger directions of the data matrix sounds plausible, there are several difficulties that have to be solved.

- First, since a preconditioner must correspond to an invertible transformation, it is not clear how to form a preconditioner based on a low rank approximation and, in particular, how should we treat the non-leading components.
- One of the main technical challenges in our work is to translate the approximation guarantees of the Lanczos method into a guarantee on the resulted average condition number. The standard measures of success for low-rank approximation are based on either Frobenius norm or spectral norm errors. As will be apparent from the analysis (see Section 5.4), such bounds do not suffice for our needs. Our analysis relies on stronger per vector error guarantees (6) due to Musco & Musco (2015).

It should be emphasized that while we use a variant of SVRG due to Xiao & Zhang (2014), we could equally use a variant of SDCA (Shalev-Shwartz, 2016) or develop such a variant for SAG or SAGA. Furthermore, while we focus on the quadratic case, we believe that our ideas can be lifted to more general setting. For example, when applied to self-concordant functions, each step of Newton's method requires the minimization of a quadratic objective. Therefore, it is natural to ask if we can benefit from applying our method for approximating the Newton step.

### 2.2. Bias-complexity tradeoff

As we mentioned above, $\lambda$ controls a tradeoff between underfitting and overfitting. In this view, we can interpret our result as follows. Assuming for simplicity that $n \geq d$ and ignoring logarithmic dependencies, we note that if

$$\lambda = \frac{k\lambda_k + \sum_{i>k}\lambda_i}{nk} , \quad (4)$$

then the runtime of our preconditioned SVRG is $\tilde{O}(ndk)$. For comparison, the runtime of (unconditioned) SVRG is $\tilde{O}(ndk)$ if

$$\lambda = \frac{\sum_{i=1}^{d}\lambda_i}{nk} . \quad (5)$$

The ratio between the RHS of (5) and (4) is the ratio given in (3). Hence, for a given "runtime budget" of order $\tilde{O}(ndk)$, we can set the regularization parameter of the preconditioned SVRG to be smaller by this ratio. Similar interpretation holds for the accelerated versions.

## 3. Related Work

**Existing algorithms and their complexities:** Since minimizing (1) is equivalent to solving the system $(C + \lambda I)w = \frac{1}{n}\sum_{i=1}^{n} y_i x_i$, standard numerical linear algebra solvers such as Gaussian elimination can be used to solve the problem in time $O(nd^2)$.

Iterative deterministic methods, such as Gradient Descent (GD), finds an $\epsilon$-approximate minimizer in time $nd\kappa\log(1/\epsilon)$, where $\kappa = \frac{\lambda_1(C+\lambda I)}{\lambda_d(C+\lambda I)}$ is the condition number of $C + \lambda I$ (see Theorem 2.1.15 in Nesterov (2004)). The Kaczmarz algorithm (Kaczmarz, 1937) has an identical complexity. Both the Conjugate Gradient (CG) method (Hestenes & Stiefel, 1952) and the Accelerated Gradient Descent (AGD) algorithm of Nesterov (1983) enjoy a better runtime of $nd\sqrt{\kappa}\log(1/\epsilon)$. In fact, CG has a more delicate analysis (see Corollary 16.7 in Vishnoi (2012)): If all but $c \in [d]$ eigenvalues of $C + \lambda I$ are contained in a range $[a, b]$, then the runtime of CG is at most $nd(c + \sqrt{b/a}\log(1/\epsilon))$. In particular, CG's runtime is at most $O(nd^2)$. Furthermore, following the interpretation of our main result in Section 2.2, we note that for a "runtime budget" of $\tilde{O}(ndk)$, we can set the regularization parameter of CG to be of order $\lambda_k/k^2$ (which is usually much greater than the RHS of (4)).

**Linear Sketching:** Several recently developed methods in numerical linear algebra are based on the so-called *sketch-and-solve* approach, which essentially suggests that given a matrix $A$, we first replace it with a smaller random matrix $AS$, and then perform the computation on $AS$ (Woodruff, 2014; Clarkson & Woodruff, 2013; Sarlos, 2006). For example, it is known that if the entries of $S$ are i.i.d. standard normal variables and $S$ has $p = \Omega(k/\epsilon)$ columns, then with high probability, the column space of $AS$ contains a $(1 + \epsilon)$ rank-$k$ approximation to $A$ with respect to the Frobenius norm. This immediately yields a fast PCA algorithm (see Section 4.1 in Woodruff (2014)).

While the above sketch-and-solve approach sounds promising for this purpose, our analysis reveals that controlling the Frobenius norm error does not suffice for our needs. We need spectral norm bounds, which are known to be more challenging (Witten & Candès, 2013). Furthermore, as mentioned above, the success of our conditioning method heavily depends on the stronger per vector error guarantees (6) obtained by Musco & Musco (2015) which are not obtained by simpler linear sketching methods.

**Sketched preconditioning:** Recently, subspace embedding methods were used to develop cheap preconditioners for linear regression with respect to the squared loss (Woodruff, 2014). Precisely, Clarkson & Woodruff (2013) considered the case $\lambda = 0$ (i.e, standard least-squares) and developed a preconditioning method that reduces the average condition number to a constant. Thereafter, they suggest applying a basic solver such as CG. The overall running time is dominated by the preconditioning process which runs in time $\tilde{O}(d^3 + nd)$. Hence, a significant improvement over standard solvers is obtained if $n \gg d$.

The main shortcoming of this method is that it does not scale well to large dimensions. Indeed, when $d$ is very large, the overhead resulted from the preconditioning process can not be afforded.

**Efficient preconditioning based on random sampling:** While we focus on reducing the dependence on the dimensionality of the data, other work investigated the gain from using only a random subset of the data points to form the conditioner (Yang et al., 2014). The theoretical gain of this approach has been established under coherence assumptions (Yang et al., 2014).

## 4. Preliminaries

### 4.1. Additional notation and definitions

Any matrix $B \in \mathbb{R}^{d\times n}$ of rank $r$ can be written in (thin) SVD form as $B = U\Sigma V^\top = \sum_{i=1}^{r}\sigma_i(B)u_i v_i^\top$. The singular values are ordered in descending order. The spectral norm of $B$ is defined by $\|B\| = \sigma_1(B)$. The spectral norm is submultiplicative, i.e., $\|AB\| \le \|A\|\|B\|$ for all $A$ and $B$. Furthermore, the spectral norm is unitary invariant, i.e., for all $A$ and $U$ such that the columns of $U$ are orthonormal, $\|UA\| = \|A\|$. For any $k \in [r]$, it is well known that the truncated SVD of $B$, $B_k := U_k\Sigma_k V_k = \sum_{i=1}^{k}\sigma_i(B)u_i v_i^\top$, is the best rank-$k$ approximation of $B$ w.r.t. the spectral norm (Trefethen & Bau III, 1997). A twice continuously differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be $\beta$-smooth if $\|\nabla^2 f(w)\| \le \beta$ for all $w$, where $\nabla^2 f(w)$ is the Hessian of $f$ at $w$. $f$ is said to be $\alpha$-strongly convex if $\lambda_d(\nabla^2 f(w)) \ge \alpha$ for all $w$. If $g$ is convex and $f$ is $\alpha$-strongly convex, then $f + g$ is $\alpha$-strongly convex.

### 4.2. Stochastic Variance Reduced Gradient (SVRG)

We consider a variant of the Stochastic Variance Reduced Gradient (SVRG) algorithm of Johnson & Zhang (2013) due to Xiao & Zhang (2014). The algorithm is an epoch-based iterative method for minimizing an average, $F(w) = \frac{1}{N}\sum_{i=1}^{N} f_i(w)$, of smooth functions. It is assumed that each $f_i : \mathbb{R}^d \to \mathbb{R}$ is convex and $\beta_i$-smooth. The entire function $F$ is assumed to be $\alpha$-strongly convex. The algorithm is detailed in Algorithm 1. Its convergence rate

**Algorithm 1** SVRG (Xiao & Zhang, 2014)

```
1: **Input:** Functions $f_1, \dots, f_n, \beta_1, \dots, \beta_n$
2: **Parameters:** $\bar{w}_0 \in \mathbb{R}^d$, $m, \eta$, $S \in \mathbb{N}$
3: **for** $s = 1, 2, \dots, S$ **do**
4:     $\bar{w} = \bar{w}_{s-1}$
5:     $\bar{v} = \nabla F(\bar{w})$
6:     $w_0 = \bar{w}$
7:     **for** $t = 1, \dots, m$ **do** {New epoch}
8:         Pick $i_t \in [N]$ with probability $q_{i_t} = \beta_{i_t} / \sum \beta_j$
9:         $v_t = (\nabla f_{i_t}(w_{t-1}) - \nabla f_{i_t}(\bar{w}))/q_{i_t} + \bar{v}$
10:        $w_t = w_{t-1} - \eta v_t$
11:    **end for**
12:    $\bar{w}_s = \frac{1}{m} \sum_{t=1}^{m} w_t$
13: **end for**
14: **Output:** the vector $\bar{w}_S$
```

depends on the averaged smoothness of the individual functions and the average condition number of $F$, defined as

$$\hat{\beta} = \frac{1}{N} \sum_{i=1}^{N} \beta_i \quad ; \quad \hat{\kappa}_F = \frac{\hat{\beta}}{\alpha} \, .$$

**Theorem 2.** *(Xiao & Zhang, 2014) Fix $\epsilon > 0$. Running SVRG (Algorithm 1) with any $w_0$, $S \geq \log((F(w_0) - \min_{w \in \mathbb{R}^d} F(w))/\epsilon)$, $m = \lceil \hat{\kappa}_F \rceil$, and $\eta = 0.1/\hat{\beta}$ yields an $\epsilon$-approximate minimizer of $F$. Furthermore, assuming that each single gradient $\nabla f_i(w)$ can be computed in time $O(d)$, the overall runtime is $O((\hat{\kappa}_F + N) d \log(\epsilon_0/\epsilon))$.*

In the original definition of SVRG (Johnson & Zhang, 2013), the indices $i_t$ are chosen uniformly at random from $[n]$, rather than proportional to $\beta_i$. As a result, the convergence rate depends on the maximal smoothness, $\max\{\beta_i\}$, rather than the average, $\hat{\beta}$. It will be apparent from our analysis (see Theorem 4) that in our case, $\max\{\beta_i\}$ is proportional to the maximum norm of any preconditioned $x_i$. Since we rely on the improved variant of Xiao & Zhang (2014), our bound depends on the average of the $\beta_i$'s, which scale with the average norm of the preconditioned $x_i$'s. To simplify the presentation, in the sequel we refer to Algorithm 1 as SVRG.

### 4.3. Randomized Block Lanczos

A randomized variant of the Block Lanczos method due to Musco & Musco (2015) is detailed[1] in Algorithm 2. Note that the matrix $\tilde{U}_k \tilde{\Sigma}_k \tilde{V}_k^\top$ forms an SVD of the matrix $\tilde{A}_k := Q(Q^\top A)_k = \tilde{U}_k \tilde{U}_k^\top A$.

**Theorem 3.** *(Musco & Musco, 2015) Consider the run of Algorithm 2 and denote $\tilde{A}_k = \tilde{U}_k \tilde{\Sigma}_k \tilde{V}_k = \sum_{i=1}^{k} \tilde{\sigma}_i \tilde{u}_i \tilde{v}_i^\top$.*

**Algorithm 2** Block Lanczos method (Musco & Musco, 2015)

```
1: **Input:** $A \in \mathbb{R}^{d \times n}, k \leq d, \epsilon' \in (0,1)$
2: $q = \Theta\left(\frac{\log(n)}{\sqrt{\epsilon}}\right), p = qk, \Pi \sim \mathcal{N}(0,1)^{n \times k}$
3: Compute $K = [A\Pi, (AA^\top)A\Pi, \dots, (AA^\top)^{q-1}A\Pi]$
4: Orthonormalize $K$'s columns to obtain $Q \in \mathbb{R}^{d \times qk}$
5: Compute the truncated SVD $(Q^\top A)_k = \tilde{W}_k \tilde{\Sigma}_k \tilde{V}_k^\top$
6: Compute $\tilde{U}_k = Q\tilde{W}_k$
7: **Output:** the matrices $\tilde{U}_k, \tilde{\Sigma}_k, \tilde{V}_k$
```

*Denote the SVD of $A$ by $A = \sum_{i=1}^{d} \sigma_i v_i u_i^\top$. The following bounds hold with probability at least $9/10$:*

$$\|A - \tilde{A}_k\| \leq (1 + \epsilon')\|A - A_k\| \leq (1 + \epsilon')\sigma_k$$

$$\forall i \in [k], \;\; |z_i^\top AA^\top z_i - u_i^\top AA^\top u_i| = |\tilde{\sigma}_i^2 - \sigma_i^2| \leq \epsilon' \sigma_{k+1}^2 \, . \quad (6)$$

*The runtime of the algorithm is $O\left(\frac{ndk\log(n)}{\sqrt{\epsilon'}} + \frac{k^2(n+d)}{\epsilon'}\right)$.*

## 5. Sketched Conditioned SVRG

In this section we develop our sketched conditioning method. By analyzing the properties of this conditioner and combining it with SVRG, we will conclude Theorem 1.

Recall that we aim at devising cheaper preconditioners that lead to a significant reduction of the condition number. Specifically, given a parameter $k \in [d]$, we will consider only preconditioners $P^{-1/2}$ for which both the computation of $P^{-1/2}$ itself and the computation of the set $\{P^{-1/2}x_i, \dots, P^{-1/2}x_n\}$ can be carried out in time $\tilde{O}(ndk)$. We will soon elaborate more on the considerations when choosing the preconditioner, but first we would like to address some important implementation issues.

### 5.1. Preconditioned regularization

In order to implement the preconditioning scheme suggested above, we should be able to find a simple form for the function $\tilde{L}$. In particular, since we would like to use SVRG, we should write $\tilde{L}$ as an average of $n$ components whose gradients can be easily computed. Denote by $\tilde{x}_i = P^{-1/2}x_i$ for all $i \in [n]$. Since for every $i \in [n]$, $((P^{-1/2}w)^\top x_i - y_i)^2 = (w^\top \tilde{x}_i - y_i)^2$, it seems natural to write $\tilde{L}(w) = L(P^{-1/2}w)$ as follows:

$$\tilde{L}(w) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{\frac{1}{2}(w^\top \tilde{x}_i - y_i)^2}_{=:\tilde{\ell}_i} + \frac{\lambda}{2}\|P^{-1/2}w\|^2 \, .$$

Assume momentarily that $\lambda = 0$. Note that the gradient of $\tilde{\ell}_i$ at any point $w$ is given by $\nabla \tilde{\ell}_i(w_t) = (w^\top \tilde{x}_i - y_i)\tilde{x}_i$.

[1]More precisely, Algorithm 2 in Musco & Musco (2015) returns the projection matrix $\tilde{U}_k \tilde{U}_k^\top$, while we also compute the SVD of $\tilde{U}_k \tilde{U}_k^\top A$. The additional runtime is negligible.

Hence, by computing all the $\tilde{x}_i$'s in advance, we are able to apply SVRG directly to the preconditioned function and computing the stochastic gradients in time $O(d)$.

When $\lambda > 0$, the computation of the gradient at some point $w$ involves the computation of $P^{-1}w$. We would like to avoid this overhead. To this end, we decompose the regularization function as follows. Denote the standard basis of $\mathbb{R}^d$ by $e_1, \ldots, e_d$. Note that the function $L$ can be rewritten as follows:

$$L(w) = \frac{1}{n+d} \sum_{i=1}^{n+d} \ell_i(w) ,$$

where $\ell_i(w) = \frac{n+d}{n} \frac{1}{2}(w^\top x_i - y_i)^2$ for $i = 1, \ldots, n$ and $\ell_{n+i}(w) = \lambda(n+d)\frac{1}{2}(w^\top e_i)^2$ for $i = 1, \ldots, d$. Finally, denoting $b_i = P^{-1/2}e_i$ for all $i$, we can rewrite the preconditioned function $\tilde{L}$ as follows:

$$\tilde{L}(w) = \frac{1}{n+d} \sum_{i=1}^{n+d} \tilde{\ell}_i(w) ,$$

where $\tilde{\ell}_i(w) = \frac{n+d}{n} \frac{1}{2}(w^\top \tilde{x}_i - y_i)^2$ for $i = 1, \ldots, n$ and $\tilde{\ell}_{n+i}(w) = \lambda(n+d)\frac{1}{2}(w^\top b_i)^2$ for $i = 1, \ldots, d$. By computing the $\tilde{x}_i$'s and the $b_i$'s in advance, we are able to apply SVRG while computing stochastic gradients in time $O(d)$.

## 5.2. The effect of conditioning

We are now in position to address the following fundamental question: How does the choice of the preconditioner, $P^{-1/2}$, affects the resulted average condition number of the function $\tilde{L}$ (4.2)? The following lemma upper bounds $\hat{\kappa}_{\tilde{L}}$ by the average condition number of the matrix $P^{-1/2}(C + \lambda I)P^{-1/2}$, which we denote by $\tilde{\kappa}$ (when the identity of the matrix $P$ is understood).

**Theorem 4.** *Let $P^{-1/2}$ be a preconditioner. Then, the average condition number of $\tilde{L}$ is upper bounded by*

$$\hat{\kappa}_{\tilde{L}} \le \tilde{\kappa} = \frac{\operatorname{tr}(P^{-1/2}(C+\lambda I)P^{-1/2})}{\lambda_d(P^{-1/2}(C+\lambda I)P^{-1/2})} .$$

The proof is in the appendix. Note that an optimal bound of $O(d)$ is attained by the whitening matrix $P^{-1/2} = (C + \lambda I)^{-1/2}$.

## 5.3. Exact sketched conditioning

Our sketched preconditioner is based on a random approximation of the best rank-$k$ approximation of the data matrix. It will be instructive to consider first a preconditioner that is based on an exact rank-$k$ approximation of the data matrix. Let $X \in \mathbb{R}^{d \times n}$ be the matrix whose $i$-th columns is $x_i$ and let $\bar{X} = n^{-1/2}X$. Denote by $\bar{X} = \sum_{i=1}^{\operatorname{rank}(\bar{X})} \sigma_i u_i v_i^\top = U\Sigma V^\top$ the SVD of $\bar{X}$ and recall that $\bar{X}_k = \sum_{i=1}^k \sigma_i u_i v_i^\top$ is the best $k$-rank approximation of $\bar{X}$. Note that $\bar{X}\bar{X}^\top = C$ and therefore $\sigma_i^2 = \lambda_i(C) = \lambda_i$. Furthermore, the left singular vectors of $\bar{X}$, $u_1, \ldots, u_k$, coincide with the $k$ leading eigenvectors of the matrix $C$. Consider the preconditioner,

$$P^{-1/2} = \sum_{i=1}^k \frac{u_i u_i^\top}{\sqrt{\lambda_i + \lambda}} + \frac{I - \sum_{i=1}^k u_i u_i^\top}{\sqrt{\lambda_k + \lambda}} ,$$

where $u_{k+1}, \ldots, u_d$ are obtained from a completion of $u_1, \ldots, u_k$ to an orthonormal basis.

**Lemma 1.** *Let $k \in [d]$ be a parameter and assume that the regularization parameter, $\lambda$, is larger than $\lambda_d$. Using the exact sketched preconditioner, we obtain*

$$\hat{\kappa}_{\tilde{L}} \le \frac{k\lambda_k + \sum_{i>k}\lambda_i}{\lambda} + d . \tag{7}$$

*Proof.* A simple calculation shows that for $i = 1, \ldots, k$,

$$\lambda_i(P^{-1/2}(C+\lambda I)P^{-1/2}) = \frac{\lambda_i + \lambda}{\lambda_i + \lambda} = 1 .$$

Similarly, for $i = k+1, \ldots, d$,

$$\lambda_i(P^{-1/2}(C+\lambda I)P^{-1/2}) = \frac{\lambda_i + \lambda}{\lambda_k + \lambda} .$$

Finally,

$$\lambda_d(P^{-1/2}(C+\lambda I)P^{-1/2}) \ge \frac{\lambda}{\lambda_k + \lambda} .$$

Combining the above with Theorem 4, we obtain that

$$\begin{aligned} \hat{\kappa}_{\tilde{L}} &\le \frac{\operatorname{tr}(P^{-1/2}(C+\lambda I)P^{-1/2})}{\lambda_d(P^{-1/2}(C+\lambda I)P^{-1/2})} \\ &\le k\frac{\lambda_k + \lambda}{\lambda} + \sum_{i=k+1}^d \frac{\lambda_i + \lambda}{\lambda} \\ &= \frac{k\lambda_k + \sum_{i>k}\lambda_i}{\lambda} + d . \end{aligned}$$

□

## 5.4. Sketched conditioning

An exact computation of the SVD of the matrix $\bar{X}$ takes $O(nd^2)$. Instead, we will use the Block Lanczos method in order to approximate the truncated SVD of $\bar{X}$. Specifically, given a parameter $k \in [d]$, we invoke the Block Lanczos method with the parameters $\bar{X}, k$ and $\epsilon' = 1/2$. Recall that the output has the form $\tilde{X}_k = \tilde{U}_k\tilde{\Sigma}_k\tilde{V}_k^\top = \sum_{i=1}^k \tilde{\sigma}_i\tilde{u}_i\tilde{v}_i^\top$. Analogously to the exact sketched preconditioner, we define our sketched preconditioner by

$$P^{-1/2} = \sum_{i=1}^k \frac{\tilde{u}_i\tilde{u}_i^\top}{\sqrt{\tilde{\sigma}_i^2 + \lambda}} + \frac{I - \sum_{i=1}^k \tilde{u}_i\tilde{u}_i^\top}{\sqrt{\tilde{\sigma}_k^2 + \lambda}} . \tag{8}$$

**Theorem 5.** *Let $k \in [d]$ be a parameter and assume that the regularization parameter, $\lambda$, is larger that $\lambda_d$. Using the sketched preconditioner defined in (8), up to a multiplicative constant, we obtain the bound (7) on the average condition number with probability at least $9/10$.*

The rest of this section is devoted to the proof of Theorem 5. We follow along the lines of the proof of Lemma 1. Up to a multiplicative constant, we derive the same upper and lower bounds on the eigenvalues of $P^{-1/2}(C + \lambda I)P^{-1/2}$.

From now on, we assume that the bounds in Theorem 3 (where $\epsilon' = 1/2$) hold. This assumption will be valid with probability of at least $9/10$. We next introduce some notation. We can rewrite $P^{-1/2} = \tilde{U}(\tilde{\Sigma}^2 + \lambda I)^{-1/2}\tilde{U}^\top$ where $\tilde{\Sigma}$ is a diagonal $d \times d$ with $\tilde{\Sigma}_{i,i} = \tilde{\sigma}_i$ if $i \le k$ and $\tilde{\Sigma}_i = \tilde{\sigma}_k$ if $i > k$. and the columns of $\tilde{U}$ are a completion of $\tilde{u}_1, \ldots, \tilde{u}_k$ to an orthonormal basis. Recall that the SVD of $\bar{X}$ is denoted by $\bar{X} = \sum_{i=1}^d \sigma_i u_i v_i^\top = U\Sigma V^\top$.

**Lemma 2.** ***(Upper bound on the leading eigenvalue)*** *We have*

$$\lambda_1(P^{-1/2}(C + \lambda I)P^{-1/2}) \le 17 .$$

*Proof.* Since $\lambda_1(P^{-1/2}(C + \lambda I)P^{-1/2}) = \|P^{-1/2}(C + \lambda I)P^{-1/2}\| = \|P^{-1/2}CP^{-1/2} + \lambda P^{-1}\|$, using the triangle inequality we have that

$$\lambda_1(P^{-1/2}(C+\lambda I)P^{-1/2}) \le \|P^{-1/2}CP^{-1/2}\| + \lambda\|P^{-1}\| .$$

By the definition of $P$ we have that $\|P^{-1}\| = \frac{1}{\tilde{\sigma}_k^2 + \lambda}$ and therefore the second summand on the right hand side of the above is at most $\frac{\lambda}{\tilde{\sigma}_k^2 + \lambda} \le 1$. As to the first summand, recall that $C = \bar{X}\bar{X}^\top$ and therefore $\|P^{-1/2}CP^{-1/2}\| = \|\bar{X}^\top P^{-1/2}\|^2$. We will show that $\|\bar{X}^\top P^{-1/2}\| \le 4$ which will imply that $\|P^{-1/2}CP^{-1/2}\| \le 16$. To do so, we first apply the triangle inequality,

$$\begin{aligned}\|\bar{X}^\top P^{-1/2}\| &= \|(\tilde{X}_k + (\bar{X} - \tilde{X}_k))^\top P^{-1/2}\| \\ &\le \|\tilde{X}_k^\top P^{-1/2}\| + \|(\bar{X} - \tilde{X}_k)^\top P^{-1/2}\| .\end{aligned}$$

Let us consider one term at the time. Recall that $\tilde{X}_k = \tilde{U}_k\tilde{\Sigma}_k\tilde{V}_k^\top$. Since $\tilde{U}_k^\top \tilde{U} \in \mathbb{R}^{k,d}$ is a diagonal matrix with ones on the diagonal, and since the spectral norm is invariant to multiplication by unitary matrices, we obtain that

$$\begin{aligned}\|\tilde{X}_k^\top P^{-1/2}\| &= \|\tilde{V}_k\tilde{\Sigma}_k\tilde{U}_k^\top\tilde{U}(\tilde{\Sigma}^2 + \lambda I)^{-1/2}\tilde{U}^\top\| \\ &= \|\tilde{\Sigma}_k\tilde{U}_k^\top\tilde{U}(\tilde{\Sigma}^2 + \lambda I)^{-1/2}\| \\ &= \max_{i\in[k]} \frac{\tilde{\sigma}_i}{\sqrt{\tilde{\sigma}_i^2 + \lambda}} \le \max_{i\in[k]} \frac{\tilde{\sigma}_i}{\tilde{\sigma}_i + \sqrt{\lambda}} \le 1 .\end{aligned}$$

Next, by the submutiplicativity of the spectral norm,

$$\|(\bar{X} - \tilde{X}_k)^\top P^{-1/2}\| \le \|\bar{X} - \tilde{X}_k\| \cdot \|P^{-1/2}\| .$$

Theorem 3 implies that $\|\bar{X} - \tilde{X}_k\| \le \frac{3}{2}\sigma_k$ and

$$\begin{aligned}\|P^{-1/2}\| &= \frac{1}{\sqrt{\tilde{\sigma}_k^2 + \lambda}} \le \frac{1}{\sqrt{\tilde{\sigma}_k^2}} \le \frac{1}{\sqrt{\sigma_k^2 - (1/2)\sigma_{k+1}^2}} \\ &\le \frac{1}{\sigma_k\sqrt{\frac{1}{2}}} = \frac{\sqrt{2}}{\sigma_k} < \frac{2}{\sigma_k} .\end{aligned}$$

Hence, $\|\bar{X} - \tilde{X}_k\| \cdot \|P^{-1/2}\| \le 3$. Combining all of the above bounds concludes our proof. □

**Lemma 3.** ***(Refined upper bound on the last $d-k$ eigenvalues)*** *For any $i \in \{k+1, \ldots, d\}$,*

$$\lambda_i\left(P^{-1/2}(C + \lambda I)P^{-1/2}\right) \le \frac{2(\lambda_i + \lambda)}{\lambda_k + \lambda} .$$

*Proof.* Using the Courant minimax principle (Bhatia, 2013), we obtain the following bound for all $i \in \{k+1, \ldots, d\}$:

$$\begin{aligned}&\lambda_i\left(P^{-1/2}(C + \lambda I)P^{-1/2}\right) \\ &= \max_{\substack{\mathcal{M}\subseteq\mathbb{R}^d:\\ \dim(\mathcal{M})=i}} \min_{\substack{x\in\mathcal{M}:\\ x\neq 0}} \frac{x^\top P^{-1/2}(C + \lambda I)P^{-1/2}x}{\|x\|^2} \\ &= \max_{\substack{\mathcal{M}\subseteq\mathbb{R}^d:\\ \dim(\mathcal{M})=i}} \min_{\substack{x\in\mathcal{M}:\\ x\neq 0}} \frac{x^\top P^{-1/2}(C + \lambda I)P^{-1/2}x}{\|P^{-1/2}x\|^2} \cdot \frac{\|P^{-1/2}x\|^2}{\|x\|^2} \\ &\le \left(\max_{\substack{\mathcal{M}\subseteq\mathbb{R}^d:\\ \dim(\mathcal{M})=i}} \min_{\substack{x\in\mathcal{M}:\\ x\neq 0}} \frac{x^\top P^{-1/2}(C + \lambda I)P^{-1/2}x}{\|P^{-1/2}x\|^2}\right) \times \\ &\quad \left(\max_{\substack{x\in\mathbb{R}^d:\\ x\neq 0}} \frac{x^\top P^{-1}x}{\|x\|^2}\right) \\ &= \lambda_i\left(C + \lambda I\right) \cdot \lambda_1(P^{-1}) = (\lambda_i + \lambda) \cdot (\tilde{\sigma}_k^2 + \lambda)^{-1} .\end{aligned}$$

Finally, using Theorem 3 we have that $\tilde{\sigma}_k^2 \ge \sigma_k^2 - \frac{1}{2}\sigma_{k+1}^2 \ge \frac{1}{2}\sigma_k^2 = \frac{1}{2}\lambda_k$ and therefore,

$$(\tilde{\sigma}_k^2 + \lambda)^{-1} \le (\tfrac{1}{2}\lambda_k + \lambda)^{-1} \le 2\,(\lambda_k + \lambda)^{-1} . \quad \square$$

**Lemma 4.** ***(Lower bound on the smallest eigenvalue)***

$$\lambda_d(P^{-1/2}CP^{-1/2}) \ge \frac{\lambda}{19(\lambda_k + \lambda)} .$$

*Proof.* Note that

$$\lambda_d(P^{-1/2}(C + \lambda I)P^{-1/2}) = \frac{1}{\|P^{1/2}(C + \lambda I)^{-1}P^{1/2}\|} , \tag{9}$$

so we can derive an upper bound on $\|P^{1/2}(C + \lambda I)^{-1}P^{1/2}\|$. Consider an arbitrary completion of

$\tilde{v}_1, \ldots, \tilde{v}_k$ to an orthonormal set, $\tilde{v}_1, \ldots, \tilde{v}_d \in \mathbb{R}^n$. Let $\tilde{V} \in \mathbb{R}^{n \times d}$ be the matrix whose $i$-th column is $\tilde{v}_i$. Since the spectral norm is unitary invariant and both $\tilde{U}$ and $\tilde{V}$ have orthonormal columns,

$$\begin{aligned}&\|P^{1/2}(C+\lambda I)^{-1}P^{1/2}\| \\ &= \|\tilde{U}(\tilde{\Sigma}^2+\lambda I)^{1/2}\tilde{U}^\top(C+\lambda I)^{-1}\tilde{U}(\tilde{\Sigma}^2+\lambda I)^{1/2}\tilde{U}^\top\| \\ &= \|\tilde{V}(\tilde{\Sigma}^2+\lambda I)^{1/2}\tilde{U}^\top(C+\lambda I)^{-1}\tilde{U}(\tilde{\Sigma}^2+\lambda I)^{1/2}\tilde{V}^\top\| .\end{aligned}$$

Denote by $\tilde{Z} = \tilde{U}(\tilde{\Sigma}^2 + \lambda I)^{1/2}\tilde{V}^\top$. By the triangle inequality and the submutiplicativity of the spectral norm,

$$\begin{aligned}\|\tilde{Z}^\top(C+\lambda I)^{-1}\tilde{Z}\| &\le \|\bar{X}^\top(C+\lambda I)^{-1}\bar{X}\| \\ &\quad + \|(\tilde{Z}-\bar{X})^\top(C+\lambda I)^{-1}(\tilde{Z}-\bar{X})\| \\ &\le \|\bar{X}^\top(C+\lambda I)^{-1}\bar{X}\| + \|\tilde{Z}-\bar{X}\|^2\|(C+\lambda I)^{-1}\| \qquad (10)\end{aligned}$$

To bound the first summand of (10), we use the unitary invariance to obtain

$$\begin{aligned}\|\bar{X}^\top(C+\lambda I)^{-1}\bar{X}\| &= \|V\Sigma U^\top U(\Sigma^2+\lambda I)^{-1}U^\top U\Sigma V^\top\| \\ &= \|\Sigma(\Sigma^2+\lambda I)^{-1}\Sigma\| = \max_i \frac{\lambda_i^2}{\lambda_i^2+\lambda} \le 1 .\end{aligned}$$

For the second summand of (10), note that $\|(C+\lambda I)^{-1}\| = \frac{1}{\lambda_d+\lambda}$ and that, using the triangle inequality,

$$\begin{aligned}\|\tilde{Z}-\bar{X}\| &= \|(\tilde{U}\tilde{\Sigma}\tilde{V}^\top - \bar{X}) + (\tilde{Z} - \tilde{U}\tilde{\Sigma}\tilde{V}^\top)\| \\ &\le \|\tilde{U}\tilde{\Sigma}\tilde{V}^\top - \bar{X}\| + \|\tilde{U}((\tilde{\Sigma}^2+\lambda I)^{1/2} - \tilde{\Sigma})\tilde{V}^\top\| .\end{aligned}$$

By using unitary invariance together with the inequality $\sqrt{\tilde{\sigma}_i^2+\lambda} - \tilde{\sigma}_i \le \sqrt{\lambda}$ (which holds for every $i$), we get

$$\|\tilde{U}((\tilde{\Sigma}^2+\lambda I)^{1/2} - \tilde{\Sigma})\tilde{V}^\top\| = \|(\tilde{\Sigma}^2+\lambda I)^{1/2} - \tilde{\Sigma}\| \le \sqrt{\lambda} .$$

Hence, using the inequality $(x+y)^2 \le 2x^2 + 2y^2$, we obtain

$$\|\tilde{Z}-\bar{X}\|^2 \le 2\|\tilde{U}\tilde{\Sigma}\tilde{V}^\top - \bar{X}\|^2 + 2\lambda .$$

We next derive an upper bound on $\|\tilde{U}\tilde{\Sigma}\tilde{V}^\top - \bar{X}\|$. Since $\tilde{U}\tilde{\Sigma}\tilde{V}^\top = \tilde{X}_k + \tilde{\sigma}_k \sum_{i=k+1}^d \tilde{u}_i\tilde{v}_i^\top$,

$$\|\tilde{U}\tilde{\Sigma}\tilde{V}^\top - \bar{X}\| \le \|\tilde{X}_k - \bar{X}\| + \tilde{\sigma}_k \left\| \sum_{i=k+1}^d \tilde{u}_i\tilde{v}_i^\top \right\| .$$

Using Theorem 3 we know that $\|\tilde{X}_k - \bar{X}\| \le 1.5\,\sigma_k$ and that $\tilde{\sigma}_k \le \sqrt{\sigma_k^2 + 0.5\,\sigma_{k+1}^2} \le 1.5\,\sigma_k$. Combining this with the fact that $\|\sum_{i=k+1}^d \tilde{u}_i\tilde{v}_i^\top\| = 1$, we obtain

$$\|\tilde{U}\tilde{\Sigma}\tilde{V}^\top - \bar{X}\| \le 3\,\sigma_k .$$

Combining the above inequalities, we obtain

$$\begin{aligned}\|P^{1/2}(C+\lambda I)^{-1}P^{1/2}\| &\le 1 + \frac{2\cdot(3\sigma_k)^2 + 2\lambda}{\lambda_d+\lambda} \\ &\le \frac{19(\lambda_k+\lambda)}{\lambda} ,\end{aligned}$$

and using (9) we conclude our proof. □

*Proof.* **(of Theorem 5)** The three last lemmas imply that the inequalities derived during the proof of Lemma 1 remain intact up to a multiplicative constant. Therefore, the bound (7) on the condition number also holds up to a multiplicative constant. This completes the proof. □

### 5.5. Sketched Preconditioned SVRG

By equipping SVRG with the sketched preconditioner (8), we obtain the Sketched Preconditioned SVRG (see Algorithm 3).

*Proof.* **(of Theorem 1)** The theorem follows from Theorem 5 and Theorem 2. □

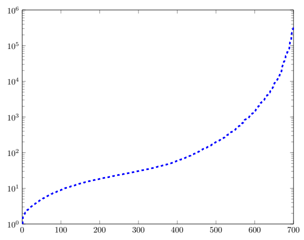
**Algorithm 3** Sketched Preconditioned SVRG

```
1:  Input: $x_1, \ldots, x_n \in \mathbb{R}^d, y_1, \ldots, y_n \in \mathbb{R}, \epsilon > 0$
2:  Parameters: $\lambda > 0, k \in [d]$
3:  Let $\bar{X} \in \mathbb{R}^{d,n}$ be the matrix whose $i$'th column is $(1/n)x_i$
4:  Run the Block Lanczos method (Algorithm 2) with the input $\bar{X}, k, \epsilon' = 1/2$ to obtain $\tilde{X}_k = \tilde{U}_k\tilde{\Sigma}_k\tilde{V}_k$
5:  Let $\tilde{u}_i$ be the columns of $\tilde{U}_k$ and $\tilde{\sigma}_i$ be the diagonal elements of $\tilde{\Sigma}_k$
6:  Form the preconditioner $P^{-1/2}$ according to (8)
7:  Compute $\tilde{x}_i = P^{-1/2}x_i, b_i = P^{-1/2}e_i$
8:  Let $\ell_i(w) = \frac{n+d}{n}\frac{1}{2}(w^\top\tilde{x}_i - y_i)^2$ for $i = 1, \ldots, n$ and $\ell_i(w) = \lambda(n+d)(w^\top b_i)^2$ for $i = n+1, \ldots, n+d$
9:  Let $\beta_i = \frac{n+d}{n}\|\tilde{x}_i\|^2$ for $i = 1, \ldots, n$ and $\beta_i = \lambda(n+d)\|b_i\|$ for $i = n+1, \ldots, n+d$. Let $\hat{\beta} = \frac{1}{n}\sum_{i=1}^{n+d}\beta_i$
10: Run SVRG (Algorithm 1)
11: Return $\hat{w} = P^{1/2}\tilde{w}$
```

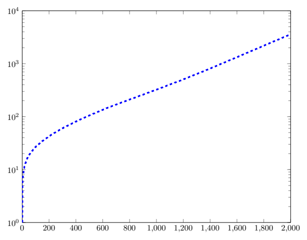
## 6. The Empirical Gain of Sketched Preconditioning

In this section we empirically demonstrate the gain of our method. We consider both regression problems and binary classifications tasks, where the square loss serves as a surrogate for the zero-one loss. We use the following datasets:

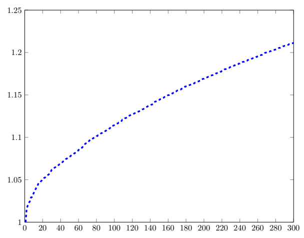
- *Synthetic*: We draw two random $5000 \times 20000$ matrices, $X^{(1)}$ and $X^{(2)}$, whose singular vectors are drawn uniformly at random and the $q$-th singular value is



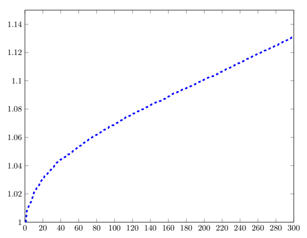
(a) MNIST dataset.



(b) CIFAR-10 dataset.



(c) RCV1 dataset.



(d) Real-sim dataset.

*Figure 1.* Plot of the ratio (3) as a function of $k$.

$1/q$ and $1/q^2$, respectively. We then normalize the columns. For each $X = X^{(j)}$, we consider a regression problem, where the labels are generated as follows: we first draw a vector $w^\star \in \mathcal{N}(0,1)^{5000}$ and then set $y_i = {w^\star}^\top X_{\cdot,i} + z_i$, where $z_i \sim \mathcal{N}(0, 0.1)$.

- *MNIST*:[2] A subset of MNIST, corresponding to the digits 4 and 7, where the task is to distinguish between the two digits. Here, $n = 12107, d = 784$.

- *RCV1*:[3] The Reuters RCV1 collection. Here, $n = 20242, d = 47236$ and we consider a standard binary document classification task.

- *CIFAR-10*:[4] Here, $n = 50000, d = 3072$. Following Frostig et al. (2015), the classification task is to distinguish between the animal categories to the automotive ones.

- *real-sim*:[5] Here, $n = 72309, d = 20958$, and we consider a standard binary document classification task.

## 6.1. Inspecting our theoretical speed-up

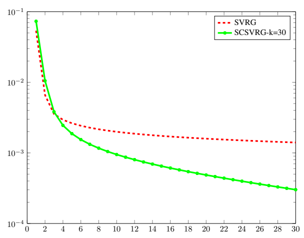
Recall that the ratio (3) quantifies our theoretical speedup. Hence, we first empirically inspect the prefixes of the corresponding quantities (as a function of $k$) for each of the datasets (see Figure 1). We can see that while in MNIST and CIFAR-10 the ratio is large for small values of $k$, in RCV1 and real-sim the ratio increases very slowly (note that for the former two datasets we use logarithmic scale).

[2]http://yann.lecun.com/exdb/mnist/

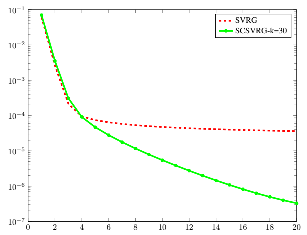
[3]https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/

[4]http://www.cs.toronto.edu/ kriz/cifar.html

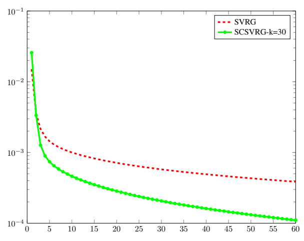
[5]https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/



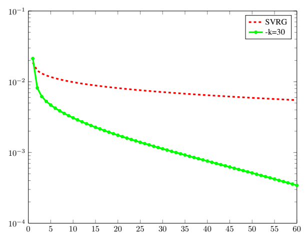
(a) Synthetic with linear decay



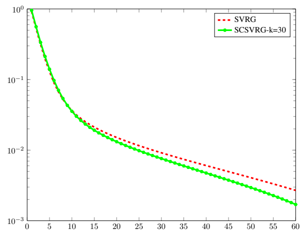
(b) Synthetic with quadratic decay



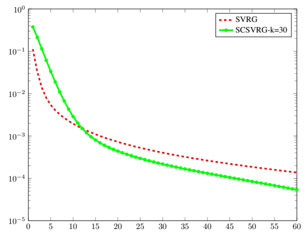
(c) MNIST dataset.



(d) CIFAR-10 dataset.



(e) RCV1 dataset.



(f) real-sim dataset.

*Figure 2.* Convergence of Sketched Preconditioned SVRG vs SVRG. The $x$-axis is the number of epochs and the $y$-axis is the suboptimality, $L(\bar{w}_t) - \min_{w \in \mathbb{R}^d} L(w)$, in logarithmic scale.

## 6.2. Empirical advantage of Sketched Preconditioned SVRG

We now evaluate Algorithm 3 and compare it to the SVRG algorithm of Xiao & Zhang (2014). To minimally affect the inherent condition number, we added only a slight amount of regularization, namely, $\lambda = 10^{-8}$. The loss used is the square loss. The step size, $\eta$, is optimally tuned for each method. Similarly to previous work on SVRG (Xiao & Zhang, 2014; Johnson & Zhang, 2013), the size of each epoch, $m$, is proportional to the number of points, $n$. We minimally preprocessed the data by average normalization: each instance vector is divided by the average $\ell_2$-norm of the instances. The number of epochs is up to 60. Note that in all cases we choose a small preconditioning parameter, namely $k = 30$, so that the preprocessing time of Algorithm 3 is negligible. There is a clear correspondence between the ratios depicted in Figure 1 and the actual speedup. In other words, the empirical results strongly affirm our theoretical results.

## Acknowledgments

We thank Edo Liberty for helpful discussions. The work is supported by ICRI-CI and by the European Research Council (TheoryDL project).

## References

Bhatia, Rajendra. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

Clarkson, Kenneth L and Woodruff, David P. Low rank approximation and regression in input sparsity time. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 81–90. ACM, 2013.

Frostig, Roy, Ge, Rong, Kakade, Sham M, and Sidford, Aaron. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. *arXiv preprint arXiv:1506.07512*, 2015.

Hestenes, Magnus Rudolph and Stiefel, Eduard. Methods of conjugate gradients for solving linear systems. NBS, 1952.

Johnson, Rie and Zhang, Tong. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pp. 315–323, 2013.

Kaczmarz, Stefan. Angenäherte auflösung von systemen linearer gleichungen. *Bulletin International de l'Académie Polonaise des Sciences et des Lettres*, 35: 355–357, 1937.

Lin, Hongzhou, Mairal, Julien, and Harchaoui, Zaid. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, pp. 3366–3374, 2015.

Musco, Cameron and Musco, Christopher. Randomized block krylov methods for stronger and faster approximate singular value decomposition. In *Advances in Neural Information Processing Systems*, pp. 1396–1404, 2015.

Nesterov, Yurii. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In *Soviet Mathematics Doklady*, volume 27, pp. 372–376, 1983.

Nesterov, Yurii. *Introductory lectures on convex optimization*, volume 87. Springer Science & Business Media, 2004.

Roux, Nicolas L, Schmidt, Mark, and Bach, Francis R. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems*, pp. 2663–2671, 2012.

Sarlos, Tamas. Improved approximation algorithms for large matrices via random projections. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pp. 143–152. IEEE, 2006.

Shalev-Shwartz, Shai. Sdca without duality, regularization, and individual convexity. *arXiv preprint arXiv:1602.01582*, 2016.

Shalev-Shwartz, Shai and Ben-David, Shai. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Shalev-Shwartz, Shai and Zhang, Tong. Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, 14(1):567–599, 2013.

Shalev-Shwartz, Shai and Zhang, Tong. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, pp. 1–41, 2014.

Trefethen, Lloyd N and Bau III, David. *Numerical linear algebra*, volume 50. Siam, 1997.

Vishnoi, Nisheeth K. Laplacian solvers and their algorithmic applications. *Theoretical Computer Science*, 8(1-2): 1–141, 2012.

Witten, Rafi and Candès, Emmanuel. Randomized algorithms for low-rank matrix factorizations: sharp performance bounds. *Algorithmica*, 72(1):264–281, 2013.

Woodruff, David P. Sketching as a tool for numerical linear algebra. *arXiv preprint arXiv:1411.4357*, 2014.

Xiao, Lin and Zhang, Tong. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

Yang, Tianbao, Jin, Rong, Zhu, Shenghuo, and Lin, Qihang. On data preconditioning for regularized loss minimization. *Machine Learning*, pp. 1–23, 2014.