Mach Learn (2016) 103:57–79
DOI 10.1007/s10994-015-5536-6

On Data Preconditioning for Regularized Loss Minimization

Tianbao Yang[1] · Rong Jin[2] · Shenghuo Zhu[3] · Qihang Lin[4]

Received: 29 March 2015 / Accepted: 25 September 2015 / Published online: 20 October 2015

Abstract In this work, we study data preconditioning, a well-known and long-existing technique, for boosting the convergence of first-order methods for regularized loss minimization in machine learning. It is well understood that the condition number of the problem, i.e., the ratio of the Lipschitz constant to the strong convexity modulus, has a harsh effect on the convergence of the first-order optimization methods. Therefore, minimizing a small regularized loss for achieving good generalization performance, yielding an ill conditioned problem, becomes the bottleneck for big data problems. We provide a theory on data preconditioning for regularized loss minimization. In particular, our analysis exhibits an appropriate data preconditioner that is similar to zero component analysis whitening. Exploiting the concepts of numerical rank and coherence, we characterize the conditions on the loss function and on the data under which data preconditioning can reduce the condition number and therefore boost the convergence for minimizing the regularized loss. To make the data preconditioning practically useful, we propose an efficient preconditioning method through random sampling. The preliminary experiments on simulated data sets and real data sets validate our theory.

Editor: Tong Zhang.

✉ Tianbao Yang
tianbao-yang@uiowa.edu

Rong Jin
rongjin@cse.msu.edu

Shenghuo Zhu
shenghuo@gmail.com

Qihang Lin
qihang-lin@uiowa.edu

1 Department of Computer Science, University of Iowa, Iowa City, IA 52242, USA

2 Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA

3 Alibaba Group, Seattle, WA, USA

4 Department of Management Sciences, University of Iowa, Iowa City, IA 52242, USA

Keywords Optimization · Preconditioning · Regularized loss · Machine learning · Convergence

1 Introduction

Many supervised machine learning tasks end up with solving the following regularized loss minimization (RLM) problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{x}_i^\top \mathbf{w}, y_i) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2, \tag{1}$$

where $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ denotes the feature representation, $y_i \in \mathcal{Y}$ denotes the supervised information, $\mathbf{w} \in \mathbb{R}^d$ represents the decision vector and $\ell(z, y)$ is a convex loss function with respect to z. Examples can be found in classification (e.g., $\ell(\mathbf{x}^\top\mathbf{w}, y) = \log(1 + \exp(-y\mathbf{x}^\top\mathbf{w}))$ for logistic regression) and regression (e.g., $\ell(\mathbf{x}^\top\mathbf{w}, y) = (1/2)(\mathbf{x}^\top\mathbf{w} - y)^2$ for least square regression).

The first-order methods that base on the first-order information (i.e., gradient) have recently become the dominant approaches for solving the optimization problem in (1), due to their light computation compared to the second-order methods (e.g., the Newton method). Because of the explosive growth of data, recently many stochastic optimization algorithms have emerged to further reduce the running time of full gradient methods (Nesterov 2004), including stochastic gradient descent (SGD) (Shamir and Zhang 2013; Shalev-Shwartz et al. 2011), stochastic average gradient (SAG) (Le Roux et al. 2012), stochastic dual coordinate ascent (SDCA) (Shalev-Shwartz and Zhang 2013; Hsieh et al. 2008), stochastic variance reduced gradient (SVRG) (Johnson and Zhang 2013). One limitation of most first-order methods is that they suffer from a poor convergence if the condition number is small. For instance, the gradient-based stochastic optimization algorithm Pegasos (Shalev-Shwartz et al. 2011) for solving Support Vector Machine (SVM) with a Lipschitz continuous loss function, has a convergence rate of $O\left(\frac{\bar{L}^2}{\lambda T}\right)$, where $\bar{L}$ is the Lipschitz constant of the loss function w.r.t $\mathbf{w}$. The convergence rate reveals that the smaller the condition number (i.e., $\bar{L}^2/\lambda$), the worse the convergence. The same phenomenon occurs in optimizing a smooth loss function. Without loss of generality, the iteration complexity—the number of iterations required for achieving an ϵ-optimal solution, of SDCA, SAG and SVRG for a L-smooth loss function (whose gradient is $\bar{L}$-Lipschitz continuous) is $O\left(\left(n + \frac{\bar{L}}{\lambda}\right)\log\left(\frac{1}{\epsilon}\right)\right)$. Although the convergence is linear for a smooth loss function, however, iteration complexity would be dominated by the condition number $\bar{L}/\lambda$ if it is substantially large [1]. As supporting evidences, many studies have found that setting λ to a very small value plays a pivotal role in achieving good generalization performance (Shalev-Shwartz and Srebro 2008; Sridharan et al. 2008), especially for data sets with a large number of examples. Moreover, some theoretical analysis indicates that the value of λ could be as small as $1/n$ in order to achieve a small generalization error (Shalev-Shwartz and Srebro 2008; Shalev-Shwartz and Zhang 2013). Therefore, it arises as an interesting question "can we design first-order optimization algorithms that have less severe and even no dependence on the large condition number"?

[1] The condition number of the problem in (1) for the Lipschitz continuous loss function is referred to $\bar{L}^2/\lambda$, and for the smooth loss function is referred to $\bar{L}/\lambda$, where $\bar{L}$ is the Lipschitz constant for the function and its gradient w.r.t $\mathbf{w}$, respectively.

While most previous works target on improving the convergence rate by achieving a better dependence on the number of iterations T, few works have revolved around mitigating the dependence on the condition number. Bach and Moulines (2013) provided a new analysis of the averaged stochastic gradient (ASG) algorithm for minimizing a smooth objective function with a constant step size. They established a convergence rate of $O(1/T)$ without suffering from the small strong convexity modulus (c.f. the definition given in Definition 2). Two recent works Needell et al. (2014) and Xiao and Zhang (2014) proposed to use importance sampling instead of random sampling in stochastic gradient methods, leading to a dependence on the averaged Lipschitz constant of the individual loss functions instead of the worst Lipschitz constant. However, the convergence rate still badly depends on $1/\lambda$.

In this paper, we explore the data preconditioning for reducing the condition number of the problem (1). In contrast to many other works, the proposed data preconditioning technique can be potentially applied together with any first-order methods to improve their convergences. Data preconditioning is a long-existing technique that was used to improve the condition number of a data matrix. In the general form, data preconditioning is to apply P^{-1} to the data, where P is a non-singular matrix. It has been employed widely in solving linear systems (Axelsson 1994). In the context of convex optimization, data preconditioning has been applied to conjugate gradient and Newton methods to improve their convergence for ill-conditioned problems (Langer 2007). However, it remains unclear how data preconditioning can be used to improve the convergence of first-order methods for minimizing a regularized empirical loss. In the context of non-convex optimization, the data preconditioning by ZCA whitening has been widely adopted in learning deep neural networks from image data to speed-up the optimization (Ranzato et al. 2010; LeCun et al. 1998), though the underlying theory is barely known. Interestingly, our analysis reveals that the proposed data preconditioner is closely related to ZCA whitening and therefore shed light on the practice widely deployed in deep learning. However, an inevitable critique on the usage of data preconditioning is the computational overhead pertaining to computing the preconditioned data. Thanks to modern cluster of computers, this computational overhead can be made as minimal as possible with parallel computations (c.f. the discussions in Sect. 4.3). We also propose a random sampling approach to efficiently compute the preconditioned data.

In summary, our contributions include: (i) we present a theory on data preconditioning for the regularized loss optimization by introducing an appropriate data preconditioner (Sect. 4); (ii) we quantify the conditions under which the data preconditioning can reduce the condition number and therefore boost the convergence of the first-order optimization methods (c.f. Eqs. (8) and (9)); (iii) we present an efficient approach for computing the preconditioned data and validate the theory by experiments (Sects. 4.3, 5).

2 Related work

We review some related work in this section. In particular, we survey some stochastic optimization algorithms that belong to the category of the first-order methods and discuss the dependence of their convergence rates on the condition number and the data. To facilitate our analysis, we decouple the dependence on the data from the condition number. Henceforth, we denote by R the upper bound of the data norm, i.e., $\|\mathbf{x}\|_2 \leq R$, and by L the Lipschitz constant of the scalar loss function $\ell(z, y)$ or its gradient $\ell'(z, y)$ with respect to z depending the smoothness of the loss function. Then the gradient w.r.t $\mathbf{w}$ of the loss function is bounded by $\|\nabla_{\mathbf{w}}\ell(\mathbf{w}^\top\mathbf{x}, y)\|_2 = \|\ell'(\mathbf{w}^\top\mathbf{x}, y)\mathbf{x}\|_2 \leq LR$ if $\ell(z, y)$ is a L-Lipschitz

continuous non-smooth function. Similarly, the second order gradient can be bounded by $\|\nabla^2_{\mathbf{w}}\ell(\mathbf{w}^\top\mathbf{x}, y)\|_2 = \|\ell''(\mathbf{w}^\top\mathbf{x}, y)\mathbf{x}\mathbf{x}^\top\|_2 \leq LR^2$ assuming $\ell(z, y)$ is a L-smooth function. As a result the condition number for a L-Lipschitz continuous scalar loss function is L^2R^2/λ and is LR^2/λ for a L-smooth loss function. In the sequel, we will refer to R, i.e., the upper bound of the data norm as the data ingredient of the condition number, and refer to L/λ or L^2/λ, i.e., the ratio of the Lipschitz constant to the strong convexity modulus as the functional ingredient of the condition number. The analysis in Sects. 4 and 4.3 will exhibit how the data preconditioning affects the two ingredients.

Stochastic gradient descent is probably the most popular algorithm in stochastic optimization. Although many variants of SGD have been developed, the simplest SGD for solving the problem (1) proceeds as:

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta_t \left(\nabla\ell(\mathbf{w}_{t-1}^\top\mathbf{x}_{i_t}, y_{i_t}) + \lambda\mathbf{w}_{t-1}\right),$$

where i_t is randomly sampled from $\{1, \ldots, n\}$ and η_t is an appropriate step size. The value of the step size η_t depends on the strong convexity modulus of the objective function. If the loss function is a Lipschitz continuous function, the value of η_t can be set to $1/(\lambda t)$ (Shamir and Zhang 2013) that yields a convergence rate of $O\left(\frac{R^2L^2}{\lambda T}\right)$ with a proper averaging scheme. It has been shown that SGD achieves the minimax optimal convergence rate for a non-smooth loss function (Shamir and Zhang 2013); however, it only yields a sub-optimal convergence for a smooth loss function (i.e., $O(1/\sqrt{T})$) in terms of T. The curse of decreasing step size is the major reason that leads to the slow convergence. On the other hand, the decreasing step size is necessary due to the large variance of the stochastic gradient when approaching the optimal solution.

Recently, there are several works dedicated to improving the convergence rate for a smooth loss function. The motivation is to reduce the variance of the stochastic gradient so as to use a constant step size like the full gradient method. We briefly mention several pieces of works. Le Roux et al. (2012) proposed a stochastic average gradient (SAG) method, which maintains an averaged stochastic gradient summing from gradients on all examples and updates a randomly selected component using the current solution. Johnson and Zhang (2013) and Zhang et al. (2013) proposed accelerated SGDs using predicative variance reduction. The key idea is to use a mix of stochastic gradients and a full gradient. The two works share a similar idea that the algorithms compute a full gradient every certain iterations and construct an unbiased stochastic gradient using the full gradient and the gradients on one example. Stochastic dual coordinate ascent (SDCA) (Shalev-Shwartz and Zhang 2013) is another stochastic optimization algorithm that enjoys a fast convergence rate for smooth loss functions. Unlike SGD types of algorithms, SDCA works on the dual variables and at each iteration it samples one instance and updates the corresponding dual variable by increasing the dual objective. It was shown in Johnson and Zhang (2013) that SDCA also achieves a variance reduction. Finally, all these algorithms have a comparable linear convergence for smooth loss functions with the iteration complexity being characterized by $O\left(\left(n + \frac{R^2L}{\lambda}\right)\log(\frac{1}{\epsilon})\right)$ [2].

While most previous works target on improving the convergence rate for a better dependence on the number of iterations T, they have innocently ignored the fact of condition number. It has been observed when the condition number is very large, SGD suffers from a strikingly slow convergence due to that the step size $1/(\lambda t)$ is too large at the beginning of the iterations. The condition number is also an obstacle that prevents the scaling-up of the

[2] The stochastic algorithm in Zhang et al. (2013) has a quadratic dependence on the condition number.

variance-reduced stochastic algorithms, especially when exploring the mini-batch technique. For instance, Shalev-Shwartz and Zhang (2013) proposed a mini-batch SDCA in which the iteration complexity can be improved from $O\left(\frac{n}{\sqrt{m}}\right)$ to $O\left(\frac{n}{m}\right)$ if the condition number is reduced from n to n/m, where m is the size of the mini-batch.

Recently, there is a resurge of interest in importance sampling for stochastic optimization methods, aiming to reduce the condition number. For example, Needell et al. (2014) analyzed SGD with importance sampling for strongly convex objective that is composed of individual smooth functions, where the sample for computing a stochastic gradient is drawn from a distribution with probabilities proportional to smoothness parameters of individual smooth functions. They showed that importance sampling can lead to a speed-up, improving the iteration complexity from a quadratic dependence on the conditioning $(L/\lambda)^2$ (where L is a bound on the smoothness and λ on the strong convexity) to a linear dependence on L/λ. Zhao and Zhang (2014) and Xiao and Zhang (2014) analyzed the effect of importance sampling for stochastic mirror descent, stochastic dual coordinate ascent and stochastic variance reduced gradient method, and showed reduction on the condition number in the iteration complexity. However, all of these works could still suffer from very small strong convexity parameter λ as in (1). Recently Bach and Moulines (2013) provided a new analysis of the averaged stochastic gradient algorithm for a smooth objective function with a constant step size. They established a convergence rate of $O(1/T)$ without suffering from the small strong convexity modulus. It has been observed by empirical studies that it could outperform SAG for solving least square regression and logistic regression. However, our experiments demonstrate that with data preconditioning the convergence of SAG can be substantially improved and better than that of Bach and Moulines (2013)'s algorithm. More discussions can be found in the end of the Sect. 4.2.

In recent years, the idea of data preconditioning has been deployed in lasso (Jia and Rohe 2012; Huang and Jojic 2011; Paul et al. 2008; Wauthier et al. 2013) via pre-multiplying the data matrix X and the response vector y by suitable matrices P_X and P_y, to improve the support recovery properties. It was also brought to our attention that in Yang et al. (2015) the authors applied data preconditioning to overdetermined ℓ_p regression problems and exploited SGD for the preconditioned problem. The big difference between our work and these works is that we place emphasis on applying data preconditioning to first-order stochastic optimization algorithms for solving the RLM problem in (1). Another remarkable difference between the present work and these works is that in our study data preconditioning only applies to the feature vector $\mathbf{x}$ not the response vector y.

We also note that data preconditioning exploited in this work is different from preconditioning in some optimization algorithms that transforms the gradient by a preconditioner matrix or an adaptive matrix (Pock and Chambolle 2011; Duchi et al. 2011). It is also different from the Newton method that multiplies the gradient by the inverse of the Hessian matrix (Boyd and Vandenberghe 2004). As a comparison, the preconditioned data can be computed offline and the computational overhead can be made as minimal as possible by using a large computer cluster with parallel computations. Unlike most previous works, we strive to improve the convergence rate from the angle of reducing the condition number. We present a theory that characterizes the conditions when the proposed data preconditioning can improve the convergence compared to the one without using data preconditioning. The contributed theory and technique act as an additional flavoring in the stochastic optimization that could improve the convergence speed.

3 Preliminaries

In this section, we briefly introduce some key definitions that are useful throughout the paper and then discuss a naive approach of applying data preconditioning for the RLM problem.

Definition 1 A function $f(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ is a L-Lipschitz continuous function w.r.t a norm $\|\cdot\|$, if

$$|f(\mathbf{x}_1) - f(\mathbf{x}_2)| \leq L\|\mathbf{x}_1 - \mathbf{x}_2\|, \quad \forall \mathbf{x}_1, \mathbf{x}_2.$$

Definition 2 A convex function $f(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ is β-strongly convex w.r.t a norm $\|\cdot\|$, if for any $\alpha \in [0, 1]$

$$f(\alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2) \leq \alpha f(\mathbf{x}_1) + (1-\alpha) f(\mathbf{x}_2) - \frac{1}{2}\alpha(1-\alpha)\beta\|\mathbf{x}_1 - \mathbf{x}_2\|^2, \quad \forall \mathbf{x}_1, \mathbf{x}_2.$$

where β is also called the strong convexity modulus of f. When $f(\mathbf{x})$ is differentiable, the strong convexity is equivalent to

$$f(\mathbf{x}_1) \geq f(\mathbf{x}_2) + \langle \nabla f(\mathbf{x}_2), \mathbf{x}_1 - \mathbf{x}_2 \rangle + \frac{\beta}{2}\|\mathbf{x}_1 - \mathbf{x}_2\|^2, \quad \forall \mathbf{x}_1, \mathbf{x}_2.$$

Definition 3 A function $f(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ is L-smooth w.r.t a norm $\|\cdot\|$, if it is differentiable and its gradient is L-Lipschitz continuous, i.e.,

$$\|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\|_* \leq L\|\mathbf{x}_1 - \mathbf{x}_2\|, \quad \forall \mathbf{x}_1, \mathbf{x}_2$$

where $\|\cdot\|_*$ denotes the dual norm of $\|\cdot\|$, or equivalently

$$f(\mathbf{x}_1) \leq f(\mathbf{x}_2) + \langle \nabla f(\mathbf{x}_2), \mathbf{x}_1 - \mathbf{x}_2 \rangle + \frac{L}{2}\|\mathbf{x}_1 - \mathbf{x}_2\|^2, \quad \forall \mathbf{x}_1, \mathbf{x}_2.$$

In the sequel, we use the standard Euclidean norm to define Lipschitz and strongly convex functions. Examples of smooth loss functions include the logistic loss $\ell(\mathbf{w}; \mathbf{x}, y) = \log(1 + \exp(-y\mathbf{w}^\top \mathbf{x}))$ and the square loss $\ell(\mathbf{w}; \mathbf{x}, y) = \frac{1}{2}(\mathbf{w}^\top \mathbf{x} - y)^2$. The ℓ_2 norm regularizer $\frac{\lambda}{2}\|\mathbf{w}\|_2^2$ is a λ-strongly convex function.

Although the proposed data preconditioning can be applied to boost any first-order methods, we will restrict our attention to the stochastic gradient methods, which share the following updates for (1):

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta_t \left(g_t(\mathbf{w}_{t-1}) + \lambda \mathbf{w}_{t-1} \right), \tag{2}$$

where $g_t(\mathbf{w}_{t-1})$ denotes a stochastic gradient of the loss that depends on the original data representation. For example, the vanilla SGD for optimizing non-smooth loss uses $g_t(\mathbf{w}_{t-1}) = \nabla \ell(\mathbf{w}_{t-1}^\top \mathbf{x}_{i_t}; y_{i_t})\mathbf{x}_{i_t}$, where i_t is randomly sampled. SAG and SVRG use a particularly designed stochastic gradient for minimizing a smooth loss.

A straightforward approach by exploring data preconditioning for the solving problem in (1) is by variable transformation. Let P be a symmetric non-singular matrix under consideration. Then we can cast the problem in (1) into:

$$\min_{\mathbf{u} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{x}_i^\top P^{-1}\mathbf{u}, y_i) + \frac{\lambda}{2}\|P^{-1}\mathbf{u}\|_2^2, \tag{3}$$

which could be implemented by preconditioning the data $\widehat{\mathbf{x}}_i = P^{-1}\mathbf{x}_i$. Applying the stochastic gradient methods to the problem above we have the following update:

$$\mathbf{u}_t = \mathbf{u}_{t-1} - \eta_t \left(g_t(\mathbf{u}_{t-1}) + \lambda P^{-2}\mathbf{u}_{t-1}\right),$$

where $g_t(\mathbf{u}_{t-1})$ denotes a stochastic gradient of the loss that depends on the transformed data representation. However, there are two difficulties limiting the applications of the technique. First, what is an appropriate data preconditioner P^{-1}? Second, at each step we need to compute $P^{-2}\mathbf{u}_{t-1}$, which might add a significant cost ($O(d^2)$ if P^{-2} is pre-computed and is a dense matrix) to each iteration. To address these issues, we present a theory in the next section. In particular, we tackle three major questions: (i) what is the appropriate data preconditioner for the first-order methods to minimize the regularized loss as in (1); (ii) under what conditions (w.r.t the data and the loss function) the data preconditioning can boost the convergence; and (iii) how to efficiently compute the preconditioned data.

4 Theory

4.1 Data preconditioning for regularized loss minimization

The first question that we are about to address is "what is the condition on the loss function in order for data preconditioning to take effect". The question turns out to be related to how we construct the preconditioner. We are inclined to give the condition first and explain it when we construct the preconditioner. To facilitate our discussion, we assume that the first argument of the loss function is bounded by r, i.e., $|z| \leq r$. We defer the discussion on the value of r to the end of this section. The condition for the loss function given below is complimentary to the property of Lipschitz continuity.

Assumption 1 The scalar loss function $\ell(z, y)$ w.r.t z satisfies $\ell''(z, y) \geq \beta$ for $|z| \leq r$ and $\beta > 0$.

Below we discuss several important loss functions used in machine learning and statistics that have such a property.

- Square loss. The square loss $\ell(z, y) = \frac{1}{2}|y - z|^2$ has been used in ridge regression and classification. It is clear that the square loss satisfies the assumption for any z and $\beta = 1$.
- Logistic loss. The logistic loss $\ell(z, y) = \log(1 + \exp(-zy))$ where $y \in \{1, -1\}$ is used in logistic regression for classification. We can compute the second order gradient by $\ell''(z, y) = \sigma(yz)(1 - \sigma(yz))$, where $\sigma(z) = 1/(1 + \exp(-z))$ is the sigmoid function. Then it is not difficult to show that when $|z| \leq r$, we have $\ell''(z, y) \geq \sigma(r)(1 - \sigma(r))$. Therefore the assumption (1) holds for $\beta(r) = \sigma(r)(1 - \sigma(r))$.
- Possion regression loss. In statistics, Poisson regression is a form of regression analysis used to model count data and contingency tables. The equivalent loss function is given by $\ell(z, y) = \exp(z) - yz$. Then $\ell''(z, y) = \exp(z) \geq \exp(-r)$ for $|z| \leq r$. Therefore the assumption (1) hold for $\beta(r) = \exp(-r)$.

It is notable that the Assumption 1 does not necessarily indicate that the entire loss $(1/n)\sum_{i=1}^n \ell(\mathbf{w}^\top \mathbf{x}_i, y_i)$ is a strongly convex function w.r.t $\mathbf{w}$ since the second order gradient, i.e., $\frac{1}{n}\sum_{i=1}^m \ell''(\mathbf{w}^\top \mathbf{x}_i, y_i)\mathbf{x}_i\mathbf{x}_i^\top$ is not necessarily lower bounded by a positive constant. Therefore the introduced condition does not change the convergence rates that we have discussed. The construction of the data preconditioner is motivated by the following observation. Given $\ell''(z, y) \geq \beta$ for any $|z| \leq r$, we can define a new loss function $\phi(z, y)$ by

$$\phi(z, y) = \ell(z, y) - \frac{\beta}{2}z^2,$$

and we can easily show that $\phi(z, y)$ is convex for $|z| \leq r$. Using $\phi(z, y)$, we can transform the problem in (1) into:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \phi(\mathbf{w}^\top \mathbf{x}_i, y_i) + \frac{\beta}{2}\mathbf{w}^\top \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\top \mathbf{w} + \frac{\lambda}{2}\|\mathbf{w}\|_2^2.$$

Let $C = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\top$ denote the sample covariance matrix. We define a smoothed covariance matrix H as

$$H = \rho I + \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\top = \rho I + C,$$

where $\rho = \lambda/\beta$. Thus, the transformed problem becomes

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \phi(\mathbf{w}^\top \mathbf{x}_i, y_i) + \frac{\beta}{2}\mathbf{w}^\top H\mathbf{w}. \tag{4}$$

Using the variable transformation $\mathbf{v} \leftarrow H^{1/2}\mathbf{w}$, the above problem is equivalent to

$$\min_{\mathbf{v}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \phi(\mathbf{v}^\top H^{-1/2}\mathbf{x}_i, y_i) + \frac{\beta}{2}\|\mathbf{v}\|_2^2. \tag{5}$$

It can be shown that the optimal value of the above preconditioned problem is equal to that of the original problem (1). As a matter of fact, so far we have constructed a data preconditioner as given by $P^{-1} = H^{-1/2}$ that transforms the original feature vector $\mathbf{x}$ into a new vector $H^{-1/2}\mathbf{x}$. It is worth noting that the data preconditioning $H^{-1/2}\mathbf{x}$ is similar to the ZCA whitening transformation, which transforms the data using the covariance matrix, i.e., $C^{-1/2}\mathbf{x}$ such that the data has identity covariance matrix. Whitening transformation has found many applications in image processing (Petrou and Bosdogianni 1999), and it is also employed in independent component analysis (Hyvärinen and Oja 2000) and optimizing deep neural networks (Ranzato et al. 2010; LeCun et al. 1998). A similar idea has been used decorrelation of the covariate/features in statistics (Mardia et al. 1979). Finally, it is notable that when original data is sparse the preconditioned data may become dense, which may increase the per-iteration cost. It would pose stronger conditions for the data preconditioning to take effect. In our experiments, we focus on dense data sets.

4.2 Condition number

Besides the data, there are two additional alterations: (i) the strong convexity modulus is changed from λ to β and (ii) the loss function becomes $\phi(z, y) = \ell(z, y) - \frac{\beta}{2}z^2$. Before discussing the convergence rates of the first-order optimization methods for solving the preconditioned problem in (5), we elaborate on how the two ingredients of the condition number are affected: (i) the functional ingredient namely the ratio of the Lipschitz constant of the loss function to the strong convexity modulus and (ii) the data ingredient namely the upper bound of the data norm. We first analyze the change of the functional ingredient as summarized in the following lemma.

Lemma 1 *If $\ell(z, y)$ is a L-Lipschitz continuous function, then $\phi(z, y)$ is $(L+\beta r)$-Lipschitz continuous for $|z| \leq r$. If $\ell(z, y)$ is a L-smooth function, then $\phi(z, y)$ is a $(L-\beta)$-smooth function.*

Proof If $\ell(z, y)$ is a L-Lipschitz continuous function, the new function $\phi(z, y)$ is a $(L+\beta r)$-Lipschitz continuous for $|z| \leq r$ because

$$\begin{aligned} |\phi(z_1, y) - \phi(z_2, y)| &\leq L|z_1 - z_2| + \frac{\beta}{2}|z_1 - z_2|^2 \\ &\leq (L + \beta r)|z_1 - z_2| \end{aligned}$$

If $\ell(z, y)$ is a L-smooth function, then the following equality holds (Nesterov 2004)

$$\langle \ell'(z_1, y) - \ell'(z_2, y), z_1 - z_2 \rangle \leq L|z_1 - z_2|^2.$$

By the definition of $\phi(z, y)$, we have

$$\langle \phi'(z_1, y) + \beta z_1 - \phi'(z_2, y) - \beta z_2, z_1 - z_2 \rangle \leq L|z_1 - z_2|^2$$

Therefore

$$\langle \phi'(z_1, y) - \phi'(z_2, y), z_1 - z_2 \rangle \leq (L - \beta)|z_1 - z_2|^2$$

which implies $\phi(z, y)$ is a $(L-\beta)$-smooth function (Nesterov 2004). □

Lemma 1 indicates that after the data preconditioning the functional ingredient becomes $(L+\beta r)^2/\beta$ for a L-Lipschitz continuous non-smooth loss function and $(L-\beta)/\beta$ for a L-smooth function. Next, we analyze the upper bound of the preconditioned data $\widehat{\mathbf{x}} = H^{-1/2}\mathbf{x}$. Noting that $\|\widehat{\mathbf{x}}\|_2^2 = \mathbf{x}^\top H^{-1}\mathbf{x}$, in what follows we will focus on bounding $\max_i \mathbf{x}_i^\top H^{-1}\mathbf{x}_i$. We first derive and discuss the bound of the expectation $\mathrm{E}_i[\mathbf{x}_i^\top H^{-1}\mathbf{x}_i]$ treating i as a random variable in $\{1, \ldots, n\}$, which is useful in proving the convergence bound of the objective in expectation. Many discussions also carry over to the upper bound for individual data. Let $\frac{1}{\sqrt{n}}X = \frac{1}{\sqrt{n}}(\mathbf{x}_1, \cdots, \mathbf{x}_n) = U\Sigma V^\top$ be the singular value decomposition of X, where $U \in \mathbb{R}^{d\times d}$, $V \in \mathbb{R}^{n\times d}$ and $\Sigma = diag(\sigma_1, \ldots, \sigma_d)$, $\sigma_1 \geq \cdots \geq \sigma_d$, then $C = U\Sigma^2 U^\top$ is the eigen-decomposition of C. Thus, we have

$$\mathrm{E}_i[\mathbf{x}_i^\top H^{-1}\mathbf{x}_i] = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^\top H^{-1}\mathbf{x}_i = tr(H^{-1}C) = \sum_{i=1}^{d}\frac{\sigma_i^2}{\sigma_i^2+\rho} \triangleq \gamma(C, \rho). \tag{6}$$

where the expectation is taken over the randomness in the index i, which is also the source of randomness in stochastic gradient descent methods. We refer to $\gamma(C, \rho)$ as the numerical rank of C with respect to ρ. The first observation is that $\gamma(C, \rho)$ is a monotonically decreasing function in terms of ρ. It is straightforward to show that if X is low rank, e.g., $rank(X) = k \ll d$, then $\gamma(C, \rho) < k$. If C is full rank, the value of $\gamma(C, \rho)$ will be affected by the decay of its eigenvalues. Bach (2013) has derived the order of $\gamma(C, \rho)$ in ρ under two different decays of the eigenvalues of C. The following proposition summarizes the order of $\gamma(C, \rho)$ under two different decays of the eigenvalues.

Proposition 1 *If the eigenvalues of C follow a polynomial decay $\sigma_i^2 = i^{-2\tau}$, $\tau \geq 1/2$, then $\gamma(C, \rho) \leq O(\rho^{-1/(2\tau)})$, and if the eigenvalues of C satisfy an exponential decay $\sigma_i^2 = e^{-\tau i}$, then $\gamma(C, \rho) \leq O\left(\log\left(\frac{1}{\rho}\right)\right)$.*

For completeness, we include the proof in the "Appendix 1". In statistics (Hastie et al. 2001), $\gamma(C, \rho)$ is also referred to as the effective degree of freedom. In order to prove high probability bounds, we have to derive the upper bound for individual $\mathbf{x}_i^\top H^{-1}\mathbf{x}_i$. To this end, we introduce the following measure to quantify the incoherence of V.

Definition 4 The generalized incoherence measure of an orthogonal matrix $V \in \mathbb{R}^{n\times d}$ w.r.t to $(\sigma_1^2, \ldots, \sigma_d^2)$ and $\rho > 0$ is

$$\mu(\rho) = \max_{1\le i\le n} \frac{n}{\gamma(C,\rho)} \sum_{j=1}^{d} \frac{\sigma_j^2}{\sigma_j^2+\rho} V_{ij}^2. \tag{7}$$

Similar to the incoherence measure introduced in the compressive sensing theory (Candes and Romberg 2007), the generalized incoherence also measures the degree to which the rows in V are correlated with the canonical bases. We can also establish the relationship between the two incoherence measures. The incoherence of an orthogonal matrix $V \in \mathbb{R}^{n\times n}$ is defined as $\mu = \max_{ij} \sqrt{n} V_{ij}$ (Candes and Romberg 2007). With simple algebra, we can show that $\mu(\rho) \le \mu^2$. Since $\mu \in [1, \sqrt{n}]$, therefore $\mu(\rho) \in [1, n]$. Given the definition of $\mu(\rho)$, we have the following lemma on the upper bound of $\mathbf{x}_i^\top H^{-1}\mathbf{x}_i$.

Lemma 2 $\mathbf{x}_i^\top H^{-1}\mathbf{x}_i \le \mu(\rho)\gamma(C,\rho), \quad i = 1, \ldots, n.$

Proof Noting the SVD of $X = \sqrt{n} U \Sigma V^\top$, we have $\mathbf{x}_i = \sqrt{n} U \Sigma V_{i,*}^\top$, where $V_{i,*}$ is the i-th row of V, we have

$$\begin{aligned}\mathbf{x}_i^\top H^{-1}\mathbf{x}_i &= n V_{i,*}\Sigma U^\top U(\Sigma+\rho I)^{-1}U^\top U\Sigma V_{i,*}^\top \\ &= n V_{i,*}\Sigma(\Sigma+\rho I)^{-1}\Sigma V_{i,*}^\top = n\sum_{j=1}^{d}\frac{\sigma_j^2}{\sigma_j^2+\rho}V_{ij}^2\end{aligned}$$

Following the definition of $\mu(\rho)$, we can complete the proof

$$\max_{1\le i\le n} \mathbf{x}_i^\top H^{-1}\mathbf{x}_i \le \mu(\rho)\gamma(C,\rho)$$

□

The theorem below states the condition number of the preconditioned problem (5).

Theorem 1 *If $\ell(z, y)$ is a L-Lipschitz continuous function satisfying the condition in Assumption 1, then the condition number of the optimization problem in (5) is bounded by $\frac{(L+\beta r)^2\mu(\rho)\gamma(C,\rho)}{\beta}$, where $\rho = \lambda/\beta$. If $\ell(z, y)$ is a L-smooth function satisfying the condition in Assumption 1, then the condition number of (5) is $\frac{(L-\beta)\mu(\rho)\gamma(C,\rho)}{\beta}$.*

Following the above theorem and previous discussions on the condition number, we have the following observations about when the data preconditioning can reduce the condition number.

Observation 1 1. *If $\ell(z, y)$ is a L-Lipschitz continuous function and*

$$\frac{\lambda(L+\beta r)^2}{\beta L^2} \le \frac{R^2}{\mu(\rho)\gamma(C,\rho)} \tag{8}$$

where r is the upper bound of predictions $z = \mathbf{w}_t^\top \mathbf{x}_i$ during optimization, then the proposed data preconditioning can reduce the condition number.

2. *If $\ell(z, y)$ is L-smooth and*

$$\frac{\lambda}{\beta} - \frac{\lambda}{L} \leq \frac{R^2}{\mu(\rho)\gamma(C, \rho)} \tag{9}$$

then the proposed data preconditioning can reduce the condition number.

Remark 1 In the above conditions [(8) and (9)], we make explicit the effect from the loss function and the data. In the right hand side, the quantity $R^2/\mu(\rho)\gamma(C, \rho)$ measures the ratio between the maximum norm of the original data and that of the preconditioned data. The left hand side depends on the property of the loss function and the value of λ. Due to the unknown value of r for non-smooth optimization, we first discuss the indications of the condition for the smooth loss function and comment on the value of r in Remark 2. Let us consider β, $L \approx \Theta(1)$ (e.g. in ridge regression or regularized least square classification) and $\lambda = \Theta(1/n)$. Therefore $\rho = \lambda/\beta = \Theta(1/n)$. The condition in (9) for the smooth loss requires the ratio between the maximum norm of the original data and that of the preconditioned data is larger than $\Theta(1/n)$. If the eigenvalues of the covariance matrix follow an *exponential decay*, then $\gamma(C, \rho) = \Theta(1)$ and the condition indicates that

$$\mu(\rho) \leq \Theta(nR^2),$$

which can be satisfied easily if $R > 1$ due to the fact $\mu(\rho) \leq n$. If the eigenvalues follow a *polynomial decay* $i^{-2\tau}$, $\tau \geq 1/2$, then $\gamma(C, \rho) \leq O(\rho^{-1/(2\tau)}) = O(n^{1/(2\tau)})$, then the condition indicates that

$$\mu(\rho) \leq O\left(n^{1-\frac{1}{2\tau}} R^2\right),$$

which means the faster the decay of the eigenvalues, the easier for the condition to be satisfied. Actually, several previous works (Talwalkar and Rostamizadeh 2010; Gittens and Mahoney 2013; Yang and Jin 2014) have studied the coherence measure and demonstrated that it is not rare to have a small coherence measure for real data sets, making the above inequality easily satisfied.

If β is a small value (e.g., in logistic regression), then the satisfaction of the condition depends on the balance between the factors λ, L, β, $\gamma(C, \rho)$, $\mu(\rho)$, R^2. In practice, if β, L is known we can always check the condition by calculating the ratio between the maximum norm of the original data and that of the preconditioned data and comparing it with $\lambda/\beta - \lambda/L$. If β is unknown, we can take a trial and error method by tuning β to achieve the best performance.

Remark 2 Next, we comment on the value of r for non-smooth optimization. It was shown in Shalev-Shwartz et al. (2011) the optimal solution $\mathbf{w}_*$ to (1) can be bounded by $\|\mathbf{w}_*\| \leq O(\frac{1}{\sqrt{\lambda}})$. Theoretically we can ensure $|z| = |\mathbf{w}^\top \mathbf{x}| \leq R/\sqrt{\lambda}$ and thus $r^2 \leq R^2/\lambda$. In the worse case $r^2 = R^2/\lambda$, the condition number of the preconditioned problem for non-smooth optimization is bounded by $O\left(\left(\frac{L^2}{\beta} + \frac{R^2}{\lambda\beta}\right)\mu(\rho)\gamma(C, \rho)\right)$. Compared to the original condition number L^2R^2/λ, there may be no improvement for convergence. In practice, $\|\mathbf{w}_*\|_2$ could be much less than $1/\sqrt{\lambda}$ and therefore $r < R/\sqrt{\lambda}$, especially when λ is very small. On the other hand, when λ is too small the step sizes $1/(\lambda t)$ of SGD on the original problem at the beginning of iterations are extremely large, making the optimization unstable. This issue can be mitigated or eliminated by data preconditioning.

Remark 3 We can also analyze the straightforward approach by solving the preconditioned problem in (3) using $P^{-1} = H^{-1/2}$. Then the problem becomes:

$$\min_{\mathbf{u}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{u}^\top H^{-1/2}\mathbf{x}_i, y_i) + \frac{\lambda}{2}\mathbf{u}^\top H^{-1}\mathbf{u}, \tag{10}$$

The bound of the data ingredient follows the same analysis. The functional ingredient is $\tilde{O}\left(\frac{L(\sigma_1^2+\rho)}{\lambda}\right)$ due to that $\lambda\mathbf{u}^\top H^{-1}\mathbf{u} \geq \lambda/(\sigma_1^2+\rho)\|\mathbf{u}\|_2^2$. If $\lambda \ll \sigma_1^2$, then the condition number of the preconditioned problem still heavily depends on $1/\lambda$. Therefore, solving the naive preconditioned problem (3) with $P^{-1} = H^{-1/2}$ may not boost the convergence, which is also verified in Sect. 5 by experiments.

Remark 4 Finally, we use the example of SAG for solving least square regression to demonstrate the benefit of data preconditioning. Similar analysis carries on to other variance reduced stochastic optimization algorithms (Johnson and Zhang 2013; Shalev-Shwartz and Zhang 2013). When $\lambda = 1/n$ the iteration complexity of SAG would be dominated by $O(R^2 n \log(1/\epsilon))$ (Schmidt et al. 2013)—tens of epochs depending on the value of R^2. However, after data preconditioning the iteration complexity becomes $O(n\log(1/\epsilon))$ if $n \geq \hat{R}^2$, where $\hat{R}$ is the upper bound of the preconditioned data, which would be just few epochs. In comparison, Bach and Moulines' algorithm (Bach and Moulines 2013) suffers from an $O(\frac{d+R^2}{\epsilon})$ iteration complexity that could be much larger than $O(n\log(1/\epsilon))$, especially when required ϵ is small and R is large. Our empirical studies in Sect. 5 indeed verify these results.

4.3 Efficient data preconditioning

Now we proceed to address the third question, i.e., how to efficiently compute the preconditioned data. The data preconditioning using $H^{-1/2}$ needs to compute the square root inverse of H times $\mathbf{x}$, which usually costs a time complexity of $O(d^3)$. On the other hand, the computation of the preconditioned data for least square regression is as expensive as computing the closed form solution, which makes data preconditioning not attractive, especially for high-dimensional data. In this section, we analyze an efficient data preconditioning by random sampling. As a compromise, we might lose some gain in convergence. The key idea is to construct the preconditioner by sampling a subset of m training data, denoted by $\widehat{\mathcal{D}} = \{\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_m\}$. Then we construct new loss functions for individual data as,

$$\psi(\mathbf{w}^\top\mathbf{x}_i, y_i) = \begin{cases} \ell(\mathbf{w}^\top\mathbf{x}_i, y_i) - \frac{\beta}{2}(\mathbf{w}^\top\mathbf{x}_i)^2, & \text{if } \mathbf{x}_i \in \widehat{\mathcal{D}} \\ \ell(\mathbf{w}^\top\mathbf{x}_i, y_i), & \text{otherwise} \end{cases}$$

We define $\hat{\beta}$ and $\hat{\rho}$ as

$$\hat{\beta} = \frac{m}{n}\beta, \qquad \hat{\rho} = \frac{n}{m}\rho = \frac{n\lambda}{m\beta} = \frac{\lambda}{\hat{\beta}} \tag{11}$$

Then we can show that the original problem is equivalent to

$$\min_{\mathbf{v}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \psi(\mathbf{v}^\top\widehat{\mathcal{H}}^{-1/2}\mathbf{x}_i, y_i) + \frac{\hat{\beta}}{2}\|\mathbf{v}\|_2^2. \tag{12}$$

where $\widehat{\mathcal{H}} = \hat{\rho}I + \frac{1}{m}\sum_{i=1}^{m}\widehat{\mathbf{x}}_i\widehat{\mathbf{x}}_i^\top$. Thus, $\widehat{\mathcal{H}}^{-1/2}\mathbf{x}_i$ defines the new preconditioned data. Below we show how to efficiently compute $\widehat{\mathcal{H}}^{-1}\mathbf{x}$. Let $\frac{1}{\sqrt{m}}\hat{X} = \hat{U}\hat{\Sigma}\hat{V}^\top$ be the SVD of $\hat{X} =$

$(\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_m)$, where $\hat{U} \in \mathbb{R}^{d\times m}$, $\hat{\Sigma} = diag(\hat{\sigma}_1, \ldots, \hat{\sigma}_m)$. Then with simple algebra $\widehat{\mathcal{H}}^{-1/2}$ can be written as

$$\widehat{\mathcal{H}}^{-1/2} = (\hat{\rho} I + \hat{U}\hat{\Sigma}^2\hat{U}^\top)^{-1/2} = \hat{\rho}^{-1/2} I - \hat{U}\hat{S}\hat{U}^\top,$$

where $\hat{S} = diag(\hat{s}_1, \ldots, \hat{s}_m)$ and $\hat{s}_i = \hat{\rho}^{-1/2} - (\hat{\sigma}_i^2 + \hat{\rho})^{-1/2}$. Then the preconditioned data $\widehat{\mathcal{H}}^{-1/2}\mathbf{x}_i$ can be calculated by $\widehat{\mathcal{H}}^{-1/2}\mathbf{x}_i = \hat{\rho}^{-1/2}\mathbf{x}_i - \hat{U}(\hat{S}(\hat{U}^\top \mathbf{x}_i))$, which costs $O(md)$ time complexity. Additionally, the time complexity for computing the SVD of $\hat{X}$ is $O(m^2 d)$. Compared with the preconditioning with full data, the above procedure of preconditioning is much more efficient. Moreover, the calculation of the preconditioned data given the SVD of $\hat{X}$ can be carried out on multiple machines to make the computational overhead as minimal as possible.

It is worth noting that the random sampling approach has been used previously to construct the stochastic Hessian (Martens 2010; Byrd et al. 2011). Here, we analyze its impact on the condition number. The same analysis about the Lipschitz constant of the loss function carries over to $\psi(z, y)$, except that $\psi(z, y)$ is at most L-smooth if $\ell(z, y)$ is L-smooth. The following theorem allows us to bound the norm of the preconditioned data using $\widehat{\mathcal{H}}$.

Theorem 2 *Let $\hat{\rho}$ be defined in (11). For any $\delta \le 1/2$, If*

$$m \ge \frac{2}{\delta^2}(\mu(\hat{\rho})\gamma(C, \hat{\rho}) + 1)(t + \log d),$$

then with a probability $1 - e^{-t}$, we have

$$\mathbf{x}_i^\top \widehat{\mathcal{H}}^{-1}\mathbf{x}_i \le (1 + 2\delta)\mu(\hat{\rho})\gamma(C, \hat{\rho}), \quad \forall i = 1, \ldots, n$$

The proof of the theorem is presented in "Appendix 2". The theorem indicates that the upper bound of the preconditioned data is only scaled up by a small constant factor with an overwhelming probability compared to that using all data points to construct the preconditioner under moderate conditions when the data matrix X has a low coherence. Before ending this section, we present a similar theorem to Theorem 1 for using the efficient data preconditioning.

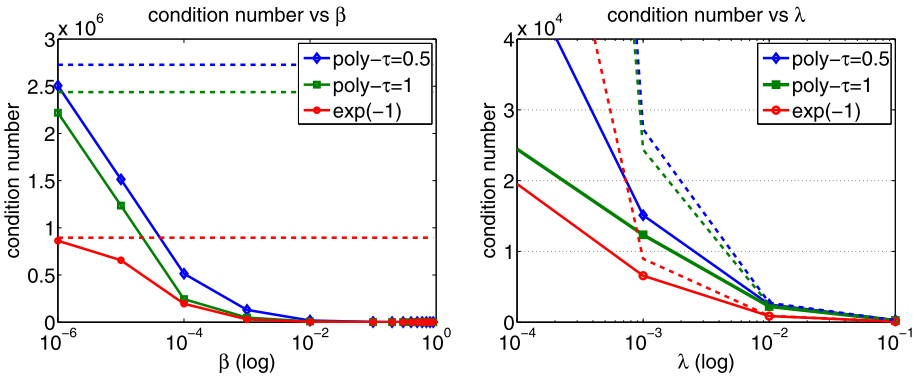
Theorem 3 *If $\ell(z, y)$ is a L-Lipschitz continuous function satisfying the condition in Assumption 1, then the condition number of the optimization problem in (12) is bounded by $\frac{(L+\beta r)^2\mu(\hat{\rho})\gamma(C,\hat{\rho})}{\hat{\beta}}$. If $\ell(z, y)$ is a L-smooth function satisfying the condition in Assumption 1, then the condition number of (12) is $\frac{L\mu(\hat{\rho})\gamma(C,\hat{\rho})}{\hat{\beta}}$.*

Thus, similar conditions can be established for the data preconditioning using $\widehat{\mathcal{H}}^{-1/2}$ to improve the convergence rate. Moreover, varying m may exhibit a tradeoff between the two ingredients understood as follows. Suppose the incoherence measure $\mu(\rho)$ is bounded by a constant. Since $\gamma(C, \hat{\rho})$ is a monotonically decreasing function w.r.t $\hat{\rho}$, therefore $\gamma(C, \hat{\rho})$ and the data ingredient $\mathbf{x}_i^\top \widehat{\mathcal{H}}^{-1}\mathbf{x}_i$ may increase as m increases. On the other hand, the functional ingredient $L/\hat{\beta}$ would decrease as m increases.

5 Experiments

5.1 Synthetic data

We first present some simulation results to verify our theory. To control the inherent data properties (i.e, numerical rank and incoherence), we generate synthetic data. We first generate



(a) fix $\lambda = 10^{-5}, L = 1$ **(b)** fix $\beta = 10^{-3}, L = 1$

Fig. 1 Synthetic data: **a** compares the condition number of the preconditioned problem (*solid lines*) with that of the original problem (*dashed lines of the same color*) by varying the value of β (a property of the loss function) and varying the decay of the eigenvalues of the sample covariance matrix (a property of the data); **b** compares the condition number by varying the value of λ (measuring the difficulty of the problem) and varying the decay of the eigenvalues

a standard Gaussian matrix $M \in \mathbb{R}^{d\times n}$ and then compute its SVD $M = USV^\top$. We use U and V as the left and right singular vectors to construct the data matrix $X \in \mathbb{R}^{d\times n}$. In this way, the incoherence measure of V is a small constant (around 5). We generate eigenvalues of C following a polynomial decay $\sigma_i^2 = i^{-2\tau}$ (poly-τ) and an exponential decay $\sigma_i^2 = \exp(-\tau i)$. Then we construct the data matrix $X = \sqrt{n}U\Sigma V^\top$, where $\Sigma = diag(\sigma_1, \ldots, \sigma_d)$.

We first plot the condition number for the problem in (1) and its data preconditioned problem in (5) using $H^{-1/2}$ by assuming the Lipschitz constant $L = 1$, varying the decay of the eigenvalues of the sample covariance matrix, and varying the values of β and λ. To this end, we generate a synthetic data with $n = 10^5, d = 100$. The curves in Fig. 1a show the condition number vs the values of β by varying the decay of the eigenvalues. It indicates that the data preconditioning can reduce the condition number for a broad range of values of β, the strong convexity modulus of the scalar loss function. The curves in Fig. 1b show a similar pattern of the condition number vs the values of λ by varying the decay of the eigenvalues. It also exhibits that the smaller the λ the larger reduction in the condition number.

Next, we present some experimental results on convergence. In our experiments we focus on two tasks namely least square regression and logistic regression, and we study two variance reduced SGDs namely stochastic average gradient (SAG) (Schmidt et al. 2013) and stochastic variance reduced SGD (SVRG) (Johnson and Zhang 2013). For SVRG, we set the step size as $0.1/\tilde{L}$, where $\tilde{L}$ is the smoothness parameter of the individual loss function plus the regularization term in terms of $\mathbf{w}$. The number of iterations for the inner loop in SVRG is set to $2n$ as suggested by the authors. For SAG, the theorem indicates the step size is less than $1/(16\tilde{L})$ while the authors have reported that using large step sizes like $1/\tilde{L}$ could yield better performances. Therefore we use $1/\tilde{L}$ as the step size unless otherwise specified. Note that we are not aiming to optimize the performances by using pre-trained initializations (Johnson and Zhang 2013) or by tuning the step sizes. Instead, the initial solution for all algorithms are set to zeros and the step sizes used in our experiments are either suggested in previous papers or have been observed to perform well in practice. In all experiments, we compare the convergence vs the number of epochs.

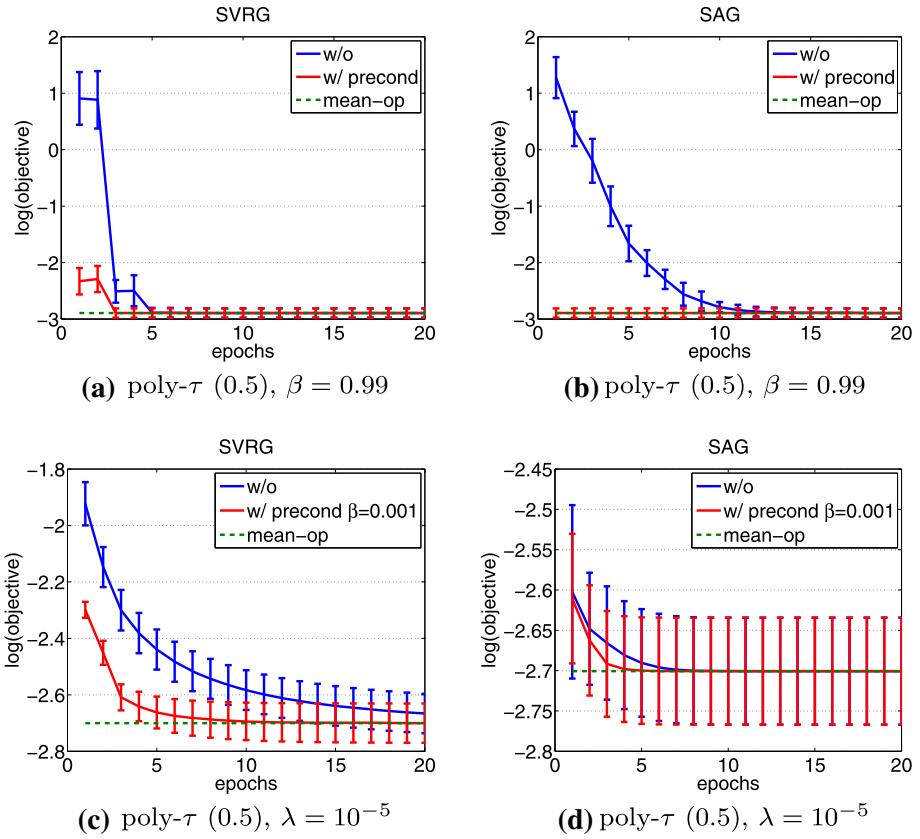
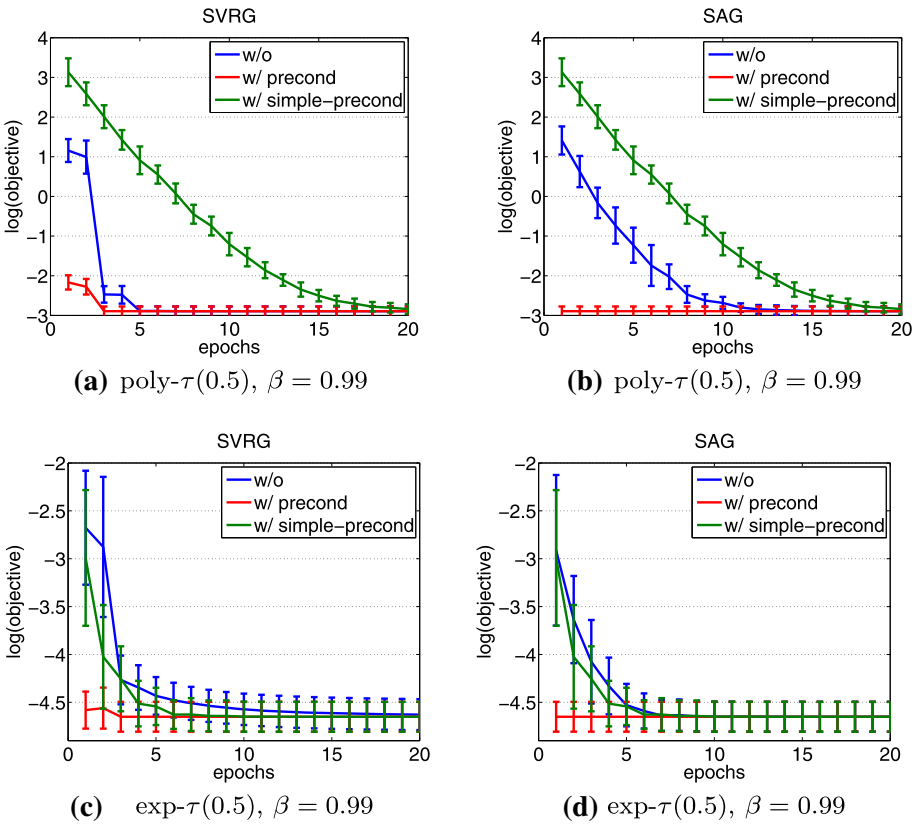


Fig. 2 Convergence of two SGD variants w/ and w/o data preconditioning for solving the least square problem (**a**, **b**) and logistic regression problem on the synthetic data with the eigenvalues following a polynomial decay. The value of λ is set to 10^{-5}. The condition numbers of the two problems are reduced from =2,727,813 and 681,953 to $c' = 1.88$, and 32,506, respectively

We generate synthetic data as described above. For least square regression, the response variable is generated by $y = \mathbf{w}^\top \mathbf{x} + \varepsilon$, where $w_i \sim \mathcal{N}(0, 100)$ and $\varepsilon \sim \mathcal{N}(0, 0.01)$. For logistic regression, the label is generated by $y = sign(\mathbf{w}^\top \mathbf{x} + \varepsilon)$. Fig. 2 shows the objective curves for minimizing the two problems by SVRG, SAG w/ and w/o data preconditioning. The results clearly demonstrate data preconditioning can significantly boost the convergence.

To further justify the proposed theory of data preconditioning, we also compare with the straightforward approach that solves the preconditioned problem in (3) with the same data preconditioner. The results are shown in Fig. 3. These results verify that using the straightforward data preconditioning may not boost the convergence.

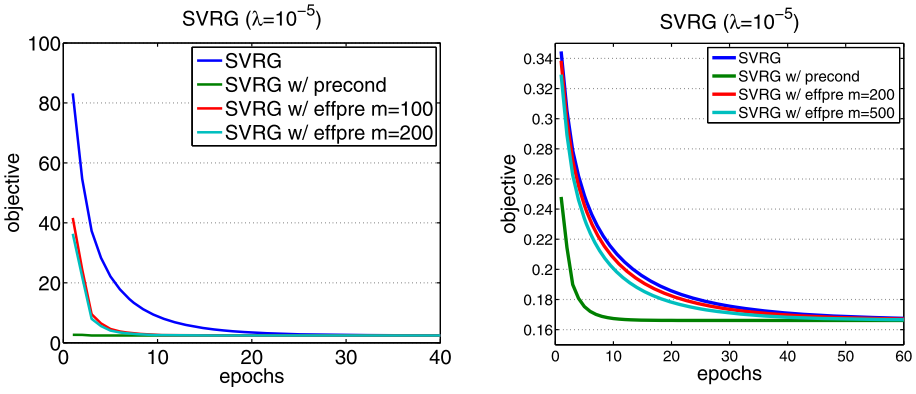
Finally, we validate the performance of the efficient data preconditioning presented in Sect. 4.3. We generate a synthetic data as before with $d = 5000$ features and with eigenvalues following the poly-0.5 decay, and plot the convergence of SVRG for solving least square regression and logistic regression with different preconditioners, including $H^{-1/2}$ and $\widehat{H}^{-1/2}$ with different values of m. The results are shown in Fig. 4, which demonstrate that using a small number m ($m = 100$ for regression and $m = 500$ for logistic regression) of training samples for constructing the data preconditioner is sufficient to gain substantial boost in the convergence.



(a) poly-$\tau(0.5)$, $\beta = 0.99$ **(b)** poly-$\tau(0.5)$, $\beta = 0.99$

(c) exp-$\tau(0.5)$, $\beta = 0.99$ **(d)** exp-$\tau(0.5)$, $\beta = 0.99$

Fig. 3 Comparison of the proposed data preconditioning with the straightforward approach by solving (3) (simple-precond) on the synthetic regression data generated with different decay of eigen-values



(a) $d = 5000, \beta = 0.99$ **(b)** $d = 5000, \beta = 0.001$

Fig. 4 Comparison of convergence of SVRG using full data and sub-sampled data for constructing the preconditioner on the synthetic data with $d = 5000$ features for regression (*left*) and logistic regression (*right*)

Table 1 The statistics of real data sets

Data set	n	d	Task
Covtype	581,012	54	Classification
MSD	463,715	90	Regression
CIFAR-10	10,000	1024	Classification
E2006-tfidf	19,395	150,350	Regression

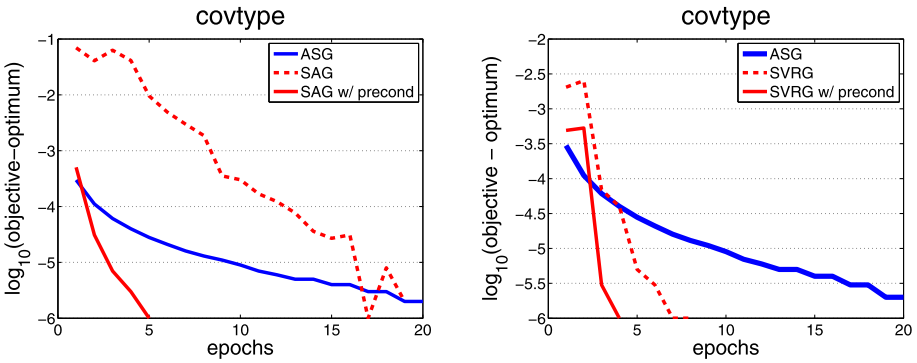


Fig. 5 Comparison of convergence on covtype. The value of λ is set to $1/n$, and the value of β is 0.01 for classification

5.2 Real data

Next, we present some experimental results on real data sets. We choose four data sets, the million songs data (MSD) (Bertin-Mahieux et al. 2011) and the E2006-tfidf data [3] (Kogan et al. 2009) for regression, and the CIFAR-10 data (Krizhevsky 2009) and the covtype data (Blackard 1998) for classification. The task on covtype is to predict the forest cover type from cartographic variables. The task on MSD is to predict the year of a song based on the audio features. Following the previous work, we map the target variable of year from 1922 to 2011 into [0, 1]. The task on CIFAR-10 is to predict the object in 32×32 RGB images. Following Krizhevsky (2009), we use the mean centered pixel values as the input. We construct a binary classification problem to classify dogs from cats with a total of 10,000 images. The task on E2006-tfidf is to predict the volatility of stock returns based on the SEC-mandated financial text report, represented by tf-idf. The size of these data sets are summarized in Table 1. We minimize regularized least square loss and regularized logistic loss for regression and classification, respectively.

The experiment results and the setup are shown in Figs. 5, 6, 7 and 8, in which we also report the convergence of Bach and Moulines' ASG algorithm (Bach and Moulines 2013) on the original problem with a step size c/R^2, where c is tuned in a range from 1 to 10. The step size for both SAG and SVRG is set to $1/\tilde{L}$. In all figures, we plot the relative objective values [4] either in log-scale or standard scale versus the epochs. For obtaining the optimal objective value, we run the fastest algorithm sufficiently long until the objective value keeps the same or is within 10^{-8} precision. On MSD and CIFAR-10, the convergence curves of optimizing

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html.

[4] the distance of the objective values to the optimal value.

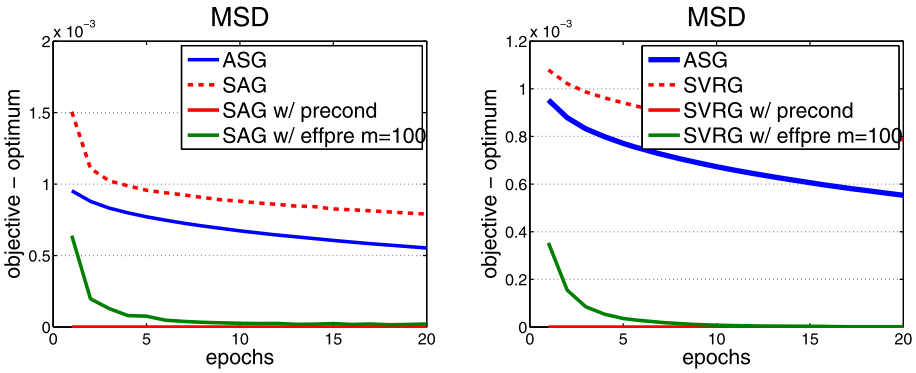


Fig. 6 Comparison of convergence on MSD. The value of λ is set to 2×10^{-6}, and the value of β is 0.99 for regression

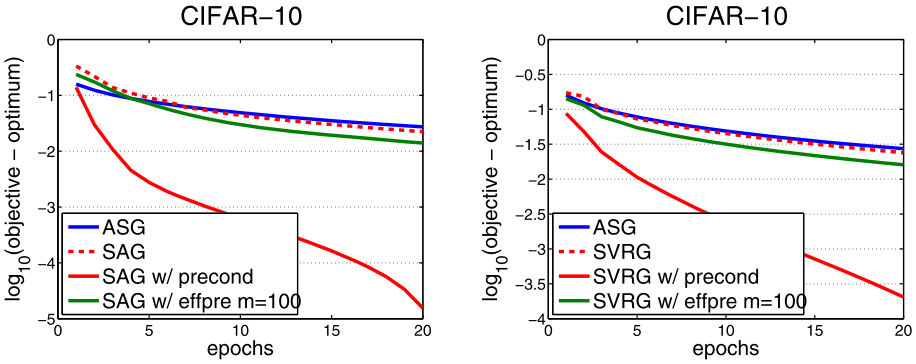


Fig. 7 Comparison of convergence on CIFAR-10. The value of λ is set to 10^{-5}, and the value of β is 0.01 for classification

the preconditioned data problem using both the full data preconditioning and the sampling based data preconditioning are plotted. On covtype, we only plot the convergence curve for optimization using the full data preconditioning, which is efficient enough. On E2006-tfidf, we only conduct optimization using the sampling based data preconditioning because the dimensionality is very large which renders the full data preconditioning very expensive. These results again demonstrate that the data preconditioning could yield significant speed-up in convergence, and the sampling based data preconditioning could be useful for high-dimensional problems.

Finally, we report some results on the running time. The computational overhead of the data preconditioning on the four data sets [5] running on Intel Xeon 3.30GHZ CPU is shown in Table 2. These computational overhead is marginal or comparable to running time per-epoch. Since the convergence on the preconditioned problem is faster than that on the original problem by tens of epochs, therefore the training on the preconditioned problem is more efficient than that on the original problem. As an example, we plot the relative objective value versus the running time on E2006-tfidf dataset in Fig. 9, where for SAG/SVRG with efficient preconditioning we count the preconditioning time at the beginning.

[5] The running time on MSD, CIFAR-10, and E2006-tfidf is for the sampling based data preconditioning and that on covtype is for the full data preconditioning.

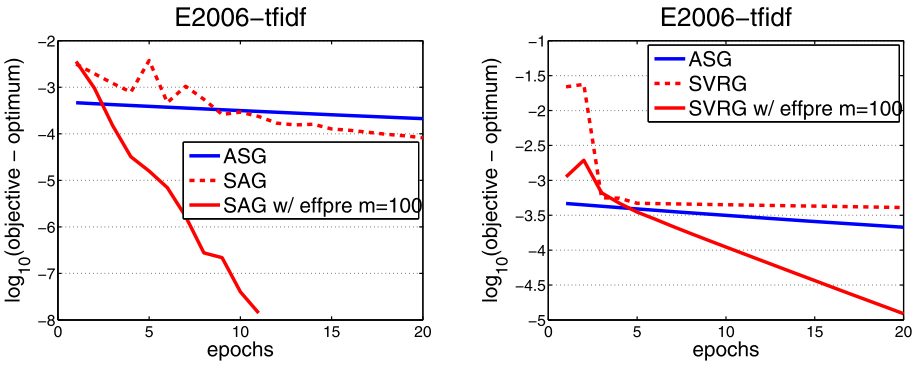


Fig. 8 Comparison of convergence on E2006-tfidf. The value of λ is set to $1/n$, and the value of β is 0.99 for regression

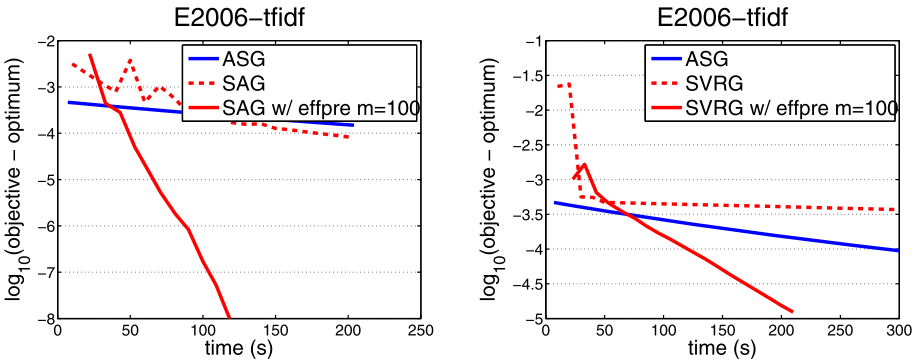


Fig. 9 comparison of convergence versus running time on E2006-tfidf. The value of λ is set to $1/n$, and the value of β is 0.99 for regression

Table 2 running time of preconditioning (p-time)

	Covtype	MSD	CIFAR-10	E2006-tfidf
p-time (s)	1.18	0.30	0.56	12

6 Conclusions

We have presented a theory of data preconditioning for boosting the convergence of first-order optimization methods for the regularized loss minimization. We characterized the conditions on the loss function and the data under which the condition number of the regularized loss minimization problem can be reduced and thus the convergence can be improved. We also presented an efficient sampling based data preconditioning which could be useful for high dimensional data, and analyzed the condition number. Our experimental results validate our theory and demonstrate the potential advantage of the data preconditioning for solving ill-conditioned regularized loss minimization problems.

Acknowledgments The authors would like to thank the anonymous reviewers for their helpful and insightful comments. T. Yang was supported in part by NSF (IIS-1463988) and NSF (IIS-1545995).

Appendix 1: Proof of Proposition 1

We first prove for the case of polynomial decay $\sigma_i^2 = i^{-2\tau}$, $\tau \geq 1/2$.

$$\begin{aligned}\sum_{i=1}^{d} \frac{\sigma_i^2}{\sigma_i^2+\rho} &= \sum_{i=1}^{d} \frac{1}{1+i^{2\tau}\rho} \leq \int_0^d \frac{1}{1+t^{2\tau}\rho} dt \\ &= \int_0^{\rho d^{2\tau}} \frac{1}{1+s} \rho^{-1/(2\tau)} s^{1/(2\tau)-1} \frac{1}{2\tau} ds \text{ (with the change of variable } s = \rho t^{2\tau}) \\ &\leq \int_0^{\infty} \frac{1}{1+s} \rho^{-1/(2\tau)} s^{1/(2\tau)-1} \frac{1}{2\tau} ds \\ &= O(\rho^{-1/(2\tau)}) \text{ (since the integral is finite)}\end{aligned}$$

For the exponential decay $\sigma_i^2 = e^{-\tau i}$, we have

$$\begin{aligned}\sum_{i=1}^{d} \frac{\sigma_i^2}{\sigma_i^2+\rho} &= \sum_{i=1}^{d} \frac{e^{-\tau i}}{e^{-\tau i}+\rho} \leq \int_0^d \frac{e^{-\tau t}}{e^{-\tau t}+\rho} dt \\ &= \frac{1}{\tau} \int_{e^{-\tau d}}^{1} \frac{s}{s+\rho} ds \text{ (with the change of variable } s = e^{-\tau t}) \\ &\leq \frac{1}{\tau} \int_0^1 \frac{s}{s+\rho} ds \leq \frac{1}{\tau} \int_0^1 \frac{1}{s+\rho} ds \\ &= \frac{1}{\tau}[\log(1+\rho) - \log(\rho)] = O\left(\log\left(\frac{1}{\rho}\right)\right)\end{aligned}$$

Appendix 2: Proof of Theorem 2

Proof Let us re-define $H = \hat{\rho} I + C$. We first show that the upper bound of the preconditioned data norm using $\widehat{\mathcal{H}}^{-1}$ is only scaled-up by a constant factor (e.g., 2) compared to that using H^{-1}. We can first bound $\mathbf{x}_i^\top \widehat{\mathcal{H}}^{-1} \mathbf{x}_i$ by $\mathbf{x}_i^\top H^{-1} \mathbf{x}_i$

$$\begin{aligned}\mathbf{x}_i^\top \widehat{\mathcal{H}}^{-1} \mathbf{x}_i &= \mathbf{x}_i^\top H^{-1/2} \left(H^{1/2} \widehat{\mathcal{H}}^{-1} H^{1/2}\right) H^{-1/2} \mathbf{x}_i \\ &\leq \lambda_{\max}\left(H^{1/2} \widehat{\mathcal{H}}^{-1} H^{1/2}\right) \mathbf{x}_i^\top H^{-1} \mathbf{x}_i, \quad i = 1, \ldots, n.\end{aligned}$$

So the crux of bounding $\mathbf{x}_i^\top \widehat{\mathcal{H}}^{-1} \mathbf{x}_i$ is to bound $\lambda_{\max}\left(H^{1/2} \widehat{\mathcal{H}}^{-1} H^{1/2}\right)$, i.e., the largest eigenvalue of $H^{1/2} \widehat{\mathcal{H}}^{-1} H^{1/2}$. To proceed the proof, we need the following Lemma. □

Lemma 3 [Tropp (2011)] *Let $\mathcal{X}$ be a finite set of PSD matrices with dimension k, and suppose that*

$$\max_{X \in \mathcal{X}} \lambda_{\max}(X) \leq B.$$

Sample $\{X_1, \ldots, X_\ell\}$ uniformly at random from $\mathcal{X}$ without replacement. Compute

$$\mu_{\max} = \ell \lambda_{\max}(\mathrm{E}[X_1]), \quad \mu_{\min} = \ell \lambda_{\min}(\mathrm{E}[X_1])$$

Then

$$\Pr\left\{\lambda_{\max}\left(\bar{X}\right) \geq (1+\delta)\mu_{\max}\right\} \leq k\left[\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right]^{\frac{\mu_{\max}}{B}}$$

$$\Pr\left\{\lambda_{\min}\left(\bar{X}\right) \leq (1-\delta)\mu_{\max}\right\} \leq k\left[\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right]^{\frac{\mu_{\max}}{B}}$$

where $\bar{X} = \sum_{i=1}^{l} X_i$.

Let us define $S = \Sigma^2 + \hat{\rho} I$ and

$$\mathcal{X} = \left\{X_i = H^{-1/2}\left(\mathbf{x}_i\mathbf{x}_i^{\top} + \hat{\rho} I\right) H^{-1/2}, i = 1, \ldots, n\right\}$$

First we show that

$$\lambda_{\max}(X_i) \leq \mu(\hat{\rho})\gamma(C, \hat{\rho}) + 1.$$

Since

$$\mu_{\max} = m\lambda_{\max}(\mathrm{E}_i[X_i]) = m$$

This can be proved by noting that

$$\lambda_{\max}(H^{-1/2}\hat{\rho} I H^{-1/2}) = \max_i \frac{\hat{\rho}}{\hat{\rho} + \sigma_i^2} \leq 1$$

$$\lambda_{\max}(H^{-1/2}\mathbf{x}_i\mathbf{x}_i^{\top} H^{-1/2}) \leq \mathbf{x}_i^{\top} H^{-1}\mathbf{x}_i \leq \mu(\hat{\rho})\gamma(C, \hat{\rho})$$

where the second inequality is due to Lemma 2 and the new definition of H. By applying the above Lemma and noting that $\bar{X} = \frac{1}{m}\sum_{i=1}^{m} X_i = H^{-1/2}\widehat{\mathcal{H}}H^{-1/2}$, we have

$$\Pr\left\{\lambda_{\min}\left(H^{-1/2}\widehat{\mathcal{H}}H^{-1/2}\right) \leq 1-\delta\right\}$$
$$\leq d\exp\left(-\frac{m}{\mu(\hat{\rho})\gamma(C, \hat{\rho}) + 1}\left[(1-\delta)\log(1-\delta) + \delta\right]\right)$$

Using the fact that

$$(1-\delta)\log(1-\delta) \geq -\delta + \frac{\delta^2}{2}$$

and by setting $m = 2(\mu(\hat{\rho})\gamma(C, \hat{\rho}) + 1)(\log d + t)/\delta^2$, we have with a probability $1 - e^{-t}$,

$$\lambda_{\min}\left(H^{-1/2}\widehat{\mathcal{H}}H^{-1/2}\right) \geq 1 - \delta$$

As a result, we have with a probability $1 - e^{-t}$,

$$\lambda_{\max}\left(H^{1/2}\widehat{\mathcal{H}}^{-1}H^{1/2}\right) \leq \frac{1}{\lambda_{\min}\left(H^{-1/2}\widehat{\mathcal{H}}H^{-1/2}\right)}$$
$$\leq \frac{1}{1-\delta} \leq 1 + 2\delta, \quad \forall \delta \leq 1/2.$$

Therefore, we have with a probability $1 - e^{-t}$ for any $\delta \leq 1/2$,

$$\mathbf{x}_i^{\top}\widehat{\mathcal{H}}^{-1}\mathbf{x}_i \leq (1+2\delta)\mu(\hat{\rho})\gamma(C, \hat{\rho}), \quad i = 1, \ldots, n$$

References

Axelsson, O. (1994). *Iterative solution methods*. New York, NY: Cambridge University Press.

Bach, F. (2013). Sharp analysis of low-rank kernel matrix approximations. In *COLT* (pp. 185–209).

Bach, F., & Moulines, E. (2013). Non-strongly-convex smooth stochastic approximation with convergence rate o(1/n). In *NIPS* (pp. 773–781).

Bertin-Mahieux, T., Ellis, D. P. W., Whitman, B., & Lamere, P. (2011). The million song dataset. In *ISMIR* (pp. 591–596).

Blackard, J. A. (1998). *Comparison of neural networks and discriminant analysis in predicting forest cover types*. Ph.D. thesis.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. New York, NY: Cambridge University Press.

Byrd, R. H., Chin, G. M., Neveitt, W., & Nocedal, J. (2011). On the use of stochastic hessian information in optimization methods for machine learning. *SIAM Journal on Optimization*, *21*, 977–995.

Candes, E. J., & Romberg, J. (2007). Sparsity and incoherence in compressive sampling. *Inverse Problems*, *23*, 969–985.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, *12*, 2121–2159.

Gittens, A., & Mahoney, M. W. (2013). Revisiting the Nystrom method for improved large-scale machine learning. *CoRR*. arXiv:1303.1849.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning. Springer series in statistics*. New York: Springer.

Hsieh, C. J., Chang, K. W., Lin, C. J., Keerthi, S. S., & Sundararajan, S. (2008). A dual coordinate descent method for large-scale linear SVM. In *ICML* (pp. 408–415)

Huang, J. C., & Jojic, N. (2011). Variable selection through correlation sifting. In *RECOMB, Lecture notes in computer science* (Vol. 6577, pp. 106–123).

Hyvärinen, A., & Oja, E. (2000). Independent component analysis: Algorithms and applications. *Neural Networks*, *13*, 411–430.

Jia, J., & Rohe, K. (2012). Preconditioning to comply with the irrepresentable condition. arXiv:1208.5584.

Johnson, R., & Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS* (pp. 315–323).

Kogan, S., Levin, D., Routledge, B. R., Sagi, J.S., & Smith, N. A. (2009). Predicting risk from financial reports with regression. In *NAACL* (pp. 272–280).

Krizhevsky, A. (2009). *Learning multiple layers of features from tiny images*. Master's thesis. Ph.D. thesis, University of Göttingen, Góttingen, Germany.

Langer, S. (2007). *Preconditioned Newton methods for Ill-posed problems*. Ph.D. thesis, University of Göttingen, Göttingen, Germany.

Le Roux, N., Schmidt, M. W., & Bach, F. (2012). A stochastic gradient method with an exponential convergence rate for finite training sets. In *NIPS* (pp. 2672–2680).

LeCun, Y., Bottou, L., Orr, G., & Müller, K. (1998). Efficient backprop. In *Neural networks: Tricks of the trade, Lecture notes in computer science*. Berlin: Springer.

Mardia, K., Kent, J., & Bibby, J. (1979). *Multivariate analysis. Probability and mathematical statistics*. London: Academic Press.

Martens, J. (2010). Deep learning via hessian-free optimization. In: *ICML* (pp. 735–742).

Needell, D., Ward, R., & Srebro, N. (2014). Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In *NIPS* (pp. 1017–1025).

Nesterov, Y. (2004). *Introductory lectures on convex optimization: A basic course. Applied optimization*. Boston: Kluwer Academic.

Paul, D., Bair, E., Hastie, T., & Tibshirani, R. (2008). Preconditioning for feature selection and regression in high-dimensional problems. *The Annals of Statistics*, *36*, 1595–1618.

Petrou, M., & Bosdogianni, P. (1999). *Image processing—The fundamentals*. New York: Wiley.

Pock, T., & Chambolle, A. (2011). Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *ICCV* (pp. 1762–1769).

Ranzato, M., Krizhevsky, A., & Hinton, G. E. (2010). Factored 3-way restricted boltzmann machines for modeling natural images. In *AISTATS* (pp. 621–628).

Schmidt, M.W., Le Roux, N., & Bach, F. (2013). Minimizing finite sums with the stochastic average gradient. *CoRR*. arXiv:1309.2388.

Shalev-Shwartz, S., Singer, Y., Srebro, N., & Cotter, A. (2011). Pegasos: Primal estimated sub-gradient solver for SVM. *Mathematical Programming*, *127*(1), 3–30.

Shalev-Shwartz, S., & Srebro, N. (2008). SVM optimization: Inverse dependence on training set size. In *ICML* (pp. 928–935).

Shalev-Shwartz, S., & Zhang, T. (2013). Accelerated mini-batch stochastic dual coordinate ascent. In *NIPS* (pp. 378–385).
Shalev-Shwartz, S., & Zhang, T. (2013). Stochastic dual coordinate ascent methods for regularized loss. *Journal of Machine Learning Research*, *14*(1), 567–599.
Shamir, O., & Zhang, T. (2013). Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *ICML* (pp. 71–79).
Sridharan, K., Shalev-Shwartz, S., & Srebro, N. (2008). Fast rates for regularized objectives. In *NIPS* (pp. 1545–1552).
Talwalkar, A., & Rostamizadeh, A. (2010). Matrix coherence and the nystrom method. In Proceedings of UAI (pp. 572–579).
Tropp, J. A. (2011). Improved analysis of the subsampled randomized Hadamard transform. *Advances in Adaptive Data Analysis*, *3*(1–2), 115–126.
Wauthier, F.L., Jojic, N., & Jordan, M. (2013). A comparative framework for preconditioned lasso algorithms. In *NIPS* (pp. 1061–1069).
Xiao, L., & Zhang, T. (2014). A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, *24*(4), 2057–2075.
Yang, J., Chow, Y. L., Re, C., & Mahoney, M. W. (2015). Weighted sgd for ℓ_p regression with randomized preconditioning. *CoRR*. arXiv:1502.03571.
Yang, T., & Jin, R. (2014). Extracting certainty from uncertainty: Transductive pairwise classification from pairwise similarities. In *Advances in neural information processing systems* (Vol. 27, pp. 262–270).
Zhang, L., Mahdavi, M., & Jin, R. (2013). Linear convergence with condition number independent access of full gradients. In *NIPS* (pp. 980–988).
Zhao, P., & Zhang, T. (2014). Stochastic optimization with importance sampling. *CoRR*. arXiv:1401.2753.