Efficient Low-Rank Stochastic Gradient Descent Methods for Solving Semidefinite Programs

Jianhui Chen
GE Global Research, San Ramon, CA 94583

Tianbao Yang and Shenghuo Zhu
NEC Labs America, Cupertino, CA 95014

Abstract

We propose a low-rank stochastic gradient descent (LR-SGD) method for solving a class of semidefinite programming (SDP) problems. LR-SGD has clear computational advantages over the standard SGD peers as its iterative projection step (a SDP problem) can be solved in an efficient manner. Specifically, LR-SGD constructs a low-rank stochastic gradient and computes an optimal solution to the projection step via analyzing the low-rank structure of its stochastic gradient. Moreover, our theoretical analysis shows the universal existence of arbitrary low-rank stochastic gradients which in turn validates the rationale of the LR-SGD method. Since LR-SGD is a SGD based method, it achieves the optimal convergence rates of the standard SGD methods. The presented experimental results demonstrate the efficiency and effectiveness of the LR-SGD method.

1 Introduction

Due to rapidly growing demands for analytic capabilities on massive data, stochastic (sub)gradient descent (SGD) based optimization methods [1] have attracted intensive research attentions, from both theorists and practitioners, in the areas of data mining and machine learning. The advantages of the SGD based methods include lightweight computation in each algorithmic iteration, drastic simplification in practical implementations, and provably rates of convergence.

Recently, semidefinite programming (SDP) has been widely employed for mathematical modeling of many data mining and machine learning applications such as distance metric learning [2], sparse covariance selection [3], matrix optimization [4], multi-task learning [5, 6], and etc.. Owing to the aforementioned advantages, SGD is a favorable method for solving those mathematical formulations, compared to traditional gradient descent (GD) based methods. Although SGD is generally very efficient, solving a mathematical formulation involving SDP constraints might still be a computational bottleneck. In particular, at each iteration of SGD, one moves an intermediate feasible solution point along the negative gradient direction towards the global optimum, usually resulting in an infeasible point. One then has to project the infeasible point into the associated positive semidefinite (PSD) cone (a SDP projection step henceforth) and compute another intermediate feasible solution for next SGD iteration. The SDP projection step is computationally expensive, as in general it needs a full-spectrum eigendecomposition on a symmetric matrix [2]. Consequently, these limitations dramatically restrict the capability of SGD in large scale data analysis.

In this paper, we propose an efficient approach to solve the SDP projection step involved in SGD. The main idea is to incorporate a low-rank stochastic gradient into the SDP projection step and utilize the special structure of the low-rank stochastic gradient for efficient computation. Specifically, we first present general procedures for constructing low-rank stochastic gradients of the objective functions and show that the low-rank constructions are naturally valid stochastic gradients; using rank-k stochastic gradients, we show that the optimal solution to the SDP projection step can be obtained via computing at most k eigenpairs of a symmetric matrix. It is worth noting that our low-rank stochastic gradient construction procedure implies the universal existence of arbitrary low-rank stochastic gradients. To the best of our knowledge, our work is the first one that employs arbitrary low-rank stochastic gradients for alleviating the computation cost in the projection into a PSD cone.

The proposed low-rank stochastic gradient can be incorporated into various SGD methods. For il-

Appearing in Proceedings of the 17^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33.

lustration, we present the details of the standard SGD method with low-rank stochastic gradients, namely, low-rank stochastic gradient descent (LR-SGD) method. Since LR-SGD is a SGD method, it achieves the same optimal rates of convergence as the standard SGD. LR-SGD can be applied for solving various machine learning formulations involving SDP constraints, for example, multi-task learning [5, 6], sparse covariance selection [7], distance metric learning [2, 8, 9], and matrix factorization [8, 9]. In our experiments, we demonstrate the effectiveness of LR-SGD on several real-world problems.

2 Related Work

We focus on discussing two recent related works. Mahdavi et al. [10] present a SGD method (called PD-SGD) which only requires one projection computation at the final iteration. The key idea is to move the domain constraint into the objective function and then solve a penalized Lagrangian function via a primal-dual stochastic gradient descent method. Albeit free of the projection computation in the intermediate iterative steps, a concomitant shortcoming of the method is that the intermediate solution (at each iteration) may not satisfy the PSD constraint from the original mathematical formulation. Hence PD-SGD may not be appropriate for certain applications, for example, in online learning, the learner may have to present to the environment a feasible solution (satisfying the PSD condition) at each iteration.

The second work is a projection-free algorithm (called OFW) proposed by Hazan and Kale [11] in online learning setting. OFW eschews the projection computation by the classical Frank-Wolfe technique [12] and efficiently finds the optimal solution to the SDP projection via solving a linear optimization problem at each iteration. This algorithm keeps the PSD property at each iteration; it is however devoted to an online setting, where at each iteration a stochastic function, instead of a stochastic gradient, is exhibited, though the latter is more widely applicable and arguably more interesting in the optimization area [13]. Another limitation of this algorithm is that it suffers from a sub-optimal convergence rate $\mathcal{O}(1/T^{1/3})$ for general convex problems, compared to the optimal convergence rate $\mathcal{O}(1/\sqrt{T})$ for standard SGD methods.

The low-rank stochastic gradients have been exploited previously for solving specific machine learning application problems. Just to name a few, the works in [14, 15, 16] have adopted an efficient rank-one update for a class of online distance metric learning problems, where the gradients of the loss functions at each iteration is simply a rank-one matrix; the work in [17] exploits low-rank stochastic gradients for efficiently computing the SVD of a matrix for solving nuclear norm regularized problems, where the loss function is a summation of functions defined on each entry of the matrix.

There are also tremendous research efforts along the direction of developing fast SDP solvers [18, 19, 20, 21, 22, 23] as well as deriving efficient optimization algorithms for solving specific application problems with SDP constraints [24, 25, 26, 27]. We do not conduct comparison with these algorithms, as they do not belong to the category of SGD methods.

Notations Denote matrices by capital bold letters. Denote by $\mathbb{S}_+ = \{\mathbf{X} \in \mathbb{R}^{d\times d} : \mathbf{X} \succeq 0\}$ a d-dimensional PSD cone. Let $\Pi_{\mathcal{D}}(\widehat{\mathbf{X}}) = \arg\min_{\mathbf{X}\in\mathcal{D}} \|\mathbf{X} - \widehat{\mathbf{X}}\|_F^2$. Let $\partial f(\mathbf{X})$ be the (sub)gradient of $f(\cdot)$ at $\mathbf{X}$. For an arbitrary symmetric matrix $\mathbf{B}$, $(\mathbf{u}_i^{\uparrow}, \lambda_i^{\uparrow})$ denotes its eigenpairs corresponding to the i-th largest eigenvalue, while $(\mathbf{u}_i^{\downarrow}, \lambda_i^{\downarrow})$ denotes its eigenpairs corresponding to the i-th smallest eigenvalue. Denote $\|\mathbf{X}\|_2 = \lambda_1^{\uparrow}, \|\mathbf{X}\|_F = \sqrt{\sum_{i=1}^d \lambda_i^2}$, and $\|\mathbf{X}\|_1 = \sum_{i=1}^d |\lambda_i|$. Denote $[x]_+ = \max(0, x)$ and $[x]_- = \min(0, x)$.

3 Gradient Descent Based Methods for Semidefinite Programming

We consider to solve the semidefinite program (SDP) [28] in the following form:

$$\begin{aligned} \min_{\mathbf{A}} \quad & f(\mathbf{A}) \\ \text{s.t.} \quad & \mathbf{A} \in \mathcal{D} = \{\mathbf{X} \,|\, \mathbf{X} \in \mathbb{S}_+, \|\mathbf{X}\| \le \lambda\}, \end{aligned} \tag{1}$$

where $f(\cdot)$ is an arbitrary convex function (not necessarily differentiable), $\|\cdot\|$ denotes a matrix norm such as the spectral norm and the Frobenius norm, and λ is a positive value (including infinity).

The formulation above is key to many machine learning tasks; the development on its efficient optimization algorithm is of broad interests in data mining and machine learning communities. For example, by combining the optimization algorithms for solving Eq. (1) with the alternating direction methods [29, 30], one can readily develop efficient methods for solving SDP problems with inequality or equality constraints in large scale settings.

To solve the SDP in Eq. (1), the gradient descent (GD) based methods [28] start from an initial solution point $\mathbf{A}_1$ and then iteratively generate intermediate solution points $\mathbf{A}_{t+1}$ $(t = 1, \ldots, T)$ by recycling the step

$$\mathbf{A}_{t+1} = \Pi_{\mathcal{D}}(\mathbf{A}_t - \gamma_t \mathcal{G}(\mathbf{A}_t)), \tag{2}$$

where $\gamma_t > 0$ denotes a step size, and $\mathcal{G}(\mathbf{A}_t) \in \partial f(\mathbf{A}_t)$ denotes a (sub)gradient of $f(\cdot)$ on $\mathbf{A}_t$. In contrast,

the stochastic gradient decent (SGD) based methods compute an optimal solution to Eq. (1) by recycling a projection step as

$$\mathbf{A}_{t+1} = \Pi_{\mathcal{D}}(\mathbf{A}_t - \gamma_t \widehat{\mathcal{G}}(\mathbf{A}_t; \omega_t)), \tag{3}$$

where ω_t is a random variable and its probability distribution P is supported on a set Ω, and $\widehat{\mathcal{G}}(\mathbf{A}; \omega)$ is a stochastic gradient that satisfies

$$\mathbb{E}_\omega \left[\widehat{\mathcal{G}}(\mathbf{A}; \omega)\right] = \int_\Omega \widehat{\mathcal{G}}(\mathbf{A}; \omega) \mathrm{d}P(\omega) \in \partial f(\mathbf{A}). \tag{4}$$

Remark It is worth noting that learning a predictor from a limited set of examples (without knowing their true distribution) is essentially a stochastic optimization problem [31]. Moreover, the SGD methods enjoy the same rate of convergence as the GD methods for optimizing general convex functions and strongly convex functions; they however generally require much less computation in each iteration and hence are suitable for large scale data analysis.

The practical efficiency of the SGD methods critically depends on the computation of the projection step (a SDP problem) in Eq. (3). The optimal solution to Eq. (3) can be routinely obtained by computing the eigendecomposition of a symmetric matrix, i.e, $\mathbf{A}_t - \gamma_t \widehat{\mathcal{G}}(\mathbf{A}_t; \omega_t)$, and then projecting the obtained eigenvalues into a convex set depending on the employed constraints. However, this procedure involves intensive computation and may be prohibitive for high dimensional problems, as the time complexity of eigendecomposition on a dense $d \times d$ matrix is $\mathcal{O}(d^3)$.

To alleviate the computation limitation, we present an efficient low-rank stochastic gradient descent (LR-SGD) method which employs a low-rank stochastic gradient and leads to time complexity $\mathcal{O}(d^2)$ in each iteration. In the following presentation, we respectively present the construction of low-rank stochastic gradients and efficient algorithms for solving the SDP projection step with low-rank stochastic gradients.

4 Low-Rank Stochastic Gradients Construction

In this section, we present the procedure of constructing low-rank stochastic gradients for Eq. (3); we defer the discussion on the efficient computation of Eq. (3) to the subsequent Section 5.

Before presenting the details of low-rank stochastic gradients construction, we present below a definition of the low-rank stochastic gradients.

Definition 1. *$\widehat{\mathcal{G}}_k(\mathbf{A}; \omega) \in \mathbb{R}^{d \times d}$ is a rank-k stochastic gradient of $f(\mathbf{A})$ if*

$$\mathbb{E}_\omega \left[\widehat{\mathcal{G}}_k(\mathbf{A}; \omega)\right] = \int_\Omega \widehat{\mathcal{G}}_k(\mathbf{A}; \omega) dP(\omega) \in \partial f(\mathbf{A}), \tag{5}$$

and Rank$(\widehat{\mathcal{G}}_k(\mathbf{A}; \omega)) = k < d$.

A straightforward approach to construct a low-rank stochastic gradient is to compute the eigendecomposion on the gradient $\mathcal{G}(\mathbf{A}) = \sum_{i=1}^d \lambda_i \mathbf{u}_i \mathbf{u}_i^\top$ and then sample k indices $\{i_1, \ldots, i_k\}$ from $\{1, 2, \cdots, d\}$ according to the distribution $P(i) = |\lambda_i| / \sum_{i=1}^d |\lambda_i|$; the rank-$k$ stochastic gradient is then obtained as $\widehat{\mathcal{G}}_k(\mathbf{A}; \omega) = (\sum_{i=1}^d |\lambda_i|/k) \sum_{j=1}^k sgn(\lambda_{i_j}) \mathbf{u}_{i_j} \mathbf{u}_{i_j}^\top$ satisfying $\mathbb{E}(\widehat{\mathcal{G}}_k(\mathbf{A}; \omega)) = \mathcal{G}(\mathbf{A})$. One limitation of this approach lies in the full spectrum eigenvalue computation, which is prohibitive for large scale data analysis.

Next, we present two efficient approaches to construct low-rank stochastic gradients.

4.1 Direct Construction

For a class of objective functions $f(\cdot)$, their gradients over $\mathbf{A}$ can be explicitly expressed as a sum of a set of rank-one matrices i.e, $\partial f(\mathbf{A}) = \sum_{i=1}^k \mathbf{u}_i \mathbf{u}_i^\top, \mathbf{u}_i \in \mathbb{R}^d$. The stochastic gradients of $f(\mathbf{A})$ can then be constructed from a selection of the available samples.

As a concrete example, we consider the problem of *Distance Metric Learning for Large Margin Nearest Neighbor Classification* (LMNN) [2] formulated as

$$\begin{aligned} \min_{\mathbf{A} \succeq 0} \ & \frac{c}{n} \sum_{i,j,l} \eta_{ij} (1 - y_{il}) \ell \left(1 + \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 - \|\mathbf{x}_i - \mathbf{x}_l\|_{\mathbf{A}}^2, 0\right) \\ & + \frac{1}{m} \sum_{i,j} \eta_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2, \end{aligned} \tag{6}$$

where $\|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{A} (\mathbf{x}_i - \mathbf{x}_j)$, $\eta_{ij} \in \{1, 0\}$ indicates whether $\mathbf{x}_j$ is a target neighbor (a nearest neighbor of the same class label) of $\mathbf{x}_i$, $y_{il} \in \{1, 0\}$ indicates whether $\mathbf{x}_i$ and $\mathbf{x}_l$ share the same class label, $\ell(z, 0) = \max(z, 0)$ denotes the hinge loss, and n, m, c denote the number of active triplets in the first term, the number of pairs in the second term, and a balancing parameter of the two terms, respectively. The goal here is to learn a distance metric that separates examples from different classes with a large margin and keeps examples from the same class closer.

A low-rank stochastic gradient of the objective function in Eq. (6) can be computed as follows. We first randomly sample a triplet (i_1, j_1, l) as well as a pair (i_2, j_2) such that the pairs $(\mathbf{x}_{i_1}, \mathbf{x}_{j_1})$ and $(\mathbf{x}_{i_2}, \mathbf{x}_{j_2})$ respectively share the same class label while $\mathbf{x}_l$ has a different class label to the pair $(\mathbf{x}_{i_1}, \mathbf{x}_{j_1})$. We then obtain a construction as $\widehat{\mathcal{G}}_k(\mathbf{A}, \omega) = c(\mathbf{x}_{i_1} - \mathbf{x}_{j_1})(\mathbf{x}_{i_1} - \mathbf{x}_{j_1})^\top - c(\mathbf{x}_{i_1} - \mathbf{x}_l)(\mathbf{x}_{i_1} - \mathbf{x}_l)^\top + (\mathbf{x}_{i_2} - \mathbf{x}_{j_2})(\mathbf{x}_{i_2} - \mathbf{x}_{j_2})^\top$ if $1 + \|\mathbf{x}_{i_1} - \mathbf{x}_{j_1}\|_{\mathbf{A}}^2 - \|\mathbf{x}_{i_1} - \mathbf{x}_l\|_{\mathbf{A}}^2 > 0$ and $\widehat{\mathcal{G}}_k(\mathbf{A}, \omega) = (\mathbf{x}_{i_2} - \mathbf{x}_{j_2})(\mathbf{x}_{i_2} - \mathbf{x}_{j_2})^\top$ otherwise. Note that the constructed $\widehat{\mathcal{G}}_k(\mathbf{A}, \omega)$ has a rank value at most 3.

4.2 Implicit Construction

For general objective functions $f(\cdot)$, their gradients may not be explicitly expressed as a sum of rank-one matrices. We present a general procedure to construct a low-rank stochastic gradient which is guaranteed to satisfy the conditions in Definition 1, and also prove that rank-2 matrices are sufficient for constructing stochastic gradients for general SDP problems.

We present below two lemmas which are main building blocks of the general procedure for constructing low-rank stochastic gradients. The first lemma shows how to construct a rank-one matrix $\widehat{\mathbf{D}}$ from an arbitrary diagonal matrix $\mathbf{D}$ so that $\mathbb{E}[\widehat{\mathbf{D}}] = \mathbf{D}$ holds .

Lemma 1. *Given any diagonal matrix $\mathbf{D} = diag(D_{11}, \cdots, D_{dd})$, let i_t be an integer sampled from the set $\{1, \ldots, d\}$ according to the distribution $\Pr(i_t) = |D_{i_t i_t}| / \sum_{i=1}^d |D_{ii}|$, and denote $\widehat{\mathbf{D}} = sgn(D_{i_t i_t}) \left(\sum_{i=1}^d |D_{ii}|\right) \mathbf{e}_{i_t} \mathbf{e}_{i_t}^\top$, where $\mathbf{e}_i \in \mathbb{R}^d$ $(i = 1, 2, \ldots, d)$ are canonical bases of $\mathbb{R}^d$. Then $\mathbb{E}[\widehat{\mathbf{D}}] = \mathbf{D}$ and $Rank(\widehat{\mathbf{D}}) = 1$.*

Proof. Since $i_t \sim \Pr(i_t)$, we can easily verify $\mathbb{E}\left[\widehat{\mathbf{D}}\right] = \sum_{j=1}^d \left(\mathrm{sgn}(D_{jj}) \left(\sum_{i=1}^d |D_{ii}|\right) \mathbf{e}_j \mathbf{e}_j^\top \Pr(j)\right) = \mathbf{D}$. This completes the proof. □

The second lemma shows how to construct a rank-one matrix $\widehat{\mathbf{P}}$ from an arbitrary off-diagonal matrix $\mathbf{P}$, so that $\mathbb{E}[\widehat{\mathbf{P}}] = \mathbf{P}$ holds.

Lemma 2. *Given an off-diagonal symmetric matrix $\mathbf{P} \in \mathbb{R}^{d \times d}$, let $\widehat{\mathbf{P}} = \frac{1}{2} \mathbf{v}^\top \mathbf{P} \mathbf{v} \mathbf{v} \mathbf{v}^\top$, where $\mathbf{v} \in \mathbb{R}^d$ and each of its entries is independently sampled from $\{1, -1\}$ with equal probability of $1/2$. Then $\mathbb{E}(\widehat{\mathbf{P}}) = \mathbf{P}$.*

Proof. We prove $\mathbb{E}(\widehat{\mathbf{P}}_{ii}) = 0$. From the definition of $\widehat{\mathbf{P}}$, we have $\mathbb{E}\left[\left(\mathbf{v}^\top \mathbf{P} \mathbf{v} \mathbf{v} \mathbf{v}^\top\right)_{ii}\right] = \mathbb{E}\left[\mathbf{v}_i \mathbf{v}_i \sum_{kl} \mathbf{v}_k \mathbf{v}_l \mathbf{P}_{kl}\right] = \mathbb{E}\left[\sum_{kl} \mathbf{v}_k \mathbf{v}_l \mathbf{P}_{kl}\right] = \sum_k \mathbf{P}_{kk} = 0$, where the second equality follows from $\mathbf{v}_i \mathbf{v}_i = 1$ and the third equality follows from $\mathbb{E}[\mathbf{v}_i \mathbf{v}_j] = 0$ if $i \neq j$.

We then prove $\mathbb{E}(\widehat{\mathbf{P}}_{ij}) = \mathbf{P}_{ij}$ for $i \neq j$. Similarly, we have $\mathbb{E}\left[\left(\mathbf{v}^\top \mathbf{P} \mathbf{v} \mathbf{v} \mathbf{v}^\top\right)_{ij}\right] = \mathbb{E}\left[\sum_{kl} \mathbf{v}_i \mathbf{v}_j \mathbf{v}_k \mathbf{v}_l \mathbf{P}_{kl}\right] = \mathbf{P}_{ij} + \mathbf{P}_{ji} = 2\mathbf{P}_{ij}$, where the second equality follows from the facts that $\mathbb{E}\left[\mathbf{v}_i \mathbf{v}_j \mathbf{v}_k \mathbf{v}_l\right] = 0$ if the indices i, j, k, and l are mutually unequal and $\mathbb{E}\left[\mathbf{v}_i \mathbf{v}_j \mathbf{v}_k \mathbf{v}_l\right] = 1$ if $i = k, j = l$ or $i = l, j = k$. Combining the results above, we complete this proof. □

From Lemmas 1 and 2, we can construct a rank-2 stochastic gradient of $f(\mathbf{A})$ as follows: (1) construct a gradient $\mathbf{G} \in \partial f(\mathbf{A})$ and split it into a diagonal matrix $\mathbf{D}$ and an off-diagonal matrix $\mathbf{P}$; (2) construct a rank-one matrix $\widehat{\mathbf{D}}$ from the diagonal component $\mathbf{D}$ as described in Lemma 1; (3) construct a rank-one matrix $\widehat{\mathbf{P}}$ from the off-diagonal component $\mathbf{P}$ as described in Lemma 2. Then a rank-2 stochastic gradient of $f(\mathbf{A})$ can be expressed as $\widehat{\mathbf{D}} + \widehat{\mathbf{P}}$. Note that given an even integer k, a rank-k stochastic gradient can be straightforwardly computed by averaging over $k/2$ repetitions of constructing $\widehat{\mathbf{D}} + \widehat{\mathbf{P}}$.

Remark The results in Lemmas 1 and 2 imply the existence of general low-rank stochastic gradients, including the rank-2 ones, for an arbitrary SDP problem.

5 Efficient Projection Computation

In this section, we discuss the efficient computation of the projection step in Eq. (3) using low-rank stochastic gradients; we show that by incorporating a rank-k stochastic gradient, the optimal solution to Eq. (3) can be obtained via computing at most k eigenpairs of a symmetric matrix.

We present detailed algorithms for using rank-2 stochastic gradients and general rank-k stochastic gradients respectively. For illustration, we consider two commonly used matrix norms, i.e, the spectral norm and the Frobenius norm for the set $\mathcal{D}$ in Eq. (1).

5.1 Projection with Rank-2 Stochastic Gradients

From Eq. (3), we denote by $\mathbf{B}_t = \mathbf{A}_t - \gamma_t \widehat{\mathcal{G}}_2(\mathbf{A}_t; \omega_t)$ the symmetric matrix (to be projected into a bounded SDP cone) from the t-th iteration of SGD. Since $\widehat{\mathcal{G}}_2(\mathbf{A}_t; \omega_t)$ is a rank-2 stochastic gradient, we can rewrite $\mathbf{B}_t$ into an explicit form as

$$\mathbf{B} = \mathbf{A} - \gamma(s_1 \mathbf{v}_1 \mathbf{v}_1^\top + s_2 \mathbf{v}_2 \mathbf{v}_2^\top), \quad \mathbf{A} \in \mathcal{D}, \tag{7}$$

where $s_1, s_2 \in \{1, 0, -1\}$ and $\gamma > 0$. Note that in Eq. (7) we suppress the iteration index t for notational simplicity. Next we present important properties of the matrix $\mathbf{B}$, as summarized in the following lemma.

Lemma 3. *Let $\mathbf{A}$, $\mathbf{B}$, s_1, s_2, and $\gamma > 0$ be defined in Eq. (7) and denote $s = s_1 + s_2$. Then*

- *if $s \in \{1, 2\}$, at most s eigenvalues of $\mathbf{B}$ are negative; moreover, if $\|\mathbf{A}\|_2 \leq \lambda$, all eigenvalues of $\mathbf{B}$ are less than or equal to λ.*
- *if $s \in \{-1, -2\}$, all eigenvalues of $\mathbf{B}$ are larger than or equal to 0; moreover, if $\|\mathbf{A}\|_2 \leq \lambda$, at most $|s|$ eigenvalues of $\mathbf{B}$ are larger than λ.*
- *if $s = 0$, at most one eigenvalue of $\mathbf{B}$ is negative; moreover, if $\|\mathbf{A}\|_2 \leq \lambda$, at most one eigenvalue of $\mathbf{B}$ is larger than λ.*

It follows from Lemma 3 that $\mathbf{B}$ in Eq. (7) has at most two negative eigenvalues and moreover $\mathbf{B}$ has at most two eigenvalues larger than λ if $\|\mathbf{A}\|_2 \leq \lambda$. Note that the lemma above can be easily proved using the standard Weyl's Inequality [32].

We present an efficient algorithm for solving Eq. (3) with the spectral norm constraint employed in $\mathcal{D}$, as summarized in the following theorem.

Theorem 4. *Let $\mathbf{A}$, $\mathbf{B}$, s_1, s_2, and γ be defined in Eq. (7) and $s = s_1 + s_2$. The optimal solution to*

$$\begin{aligned} \min_{\widehat{\mathbf{A}}} \quad & \tfrac{1}{2}\|\widehat{\mathbf{A}} - \mathbf{B}\|_F^2 \\ s.t. \quad & \widehat{\mathbf{A}} \in \mathbb{S}_+,\ \|\widehat{\mathbf{A}}\|_2 \leq \lambda \end{aligned} \tag{8}$$

is given by Rank-2-Thresholding $(\mathbf{B}, \lambda, s)$ in Algorithm 1.

Proof. Denote the Lagrangian function associated with Eq. (8) as $\mathcal{L}(\widehat{\mathbf{A}}, \mathbf{Z}_1, \mathbf{Z}_2) = \frac{1}{2}\|\widehat{\mathbf{A}} - \mathbf{B}\|_F^2 - \mathrm{tr}(\widehat{\mathbf{A}}\mathbf{Z}_1) + \mathrm{tr}((\widehat{\mathbf{A}} - \lambda\mathbf{I})\mathbf{Z}_2)$. Let $\mathbf{A}^*$ and $\{\mathbf{Z}_1^*, \mathbf{Z}_2^*\}$ be the optimal primal and dual variables to Eq. (8), respectively. The KKT conditions to Eq. (8) can be expressed as

$$\begin{aligned} &\mathbf{A}^* = \mathbf{B} + \mathbf{Z}_1^* - \mathbf{Z}_2^* \\ &\mathrm{tr}(\mathbf{A}^*\mathbf{Z}_1^*) = 0, \quad \mathrm{tr}((\mathbf{A}^* - \lambda\mathbf{I})\mathbf{Z}_2^*) = 0 \\ &\mathbf{Z}_1^* \in \mathbb{S}_+, \quad \mathbf{Z}_2^* \in \mathbb{S}_+. \end{aligned} \tag{9}$$

We compute the optimal solution to Eq. (8) by considering the following three cases.

Case 1: $\mathbf{s} \in \{-\mathbf{1}, -\mathbf{2}\}$ It follows from Lemma 3 that all eigenvalues of $\mathbf{B}$ are non-negative. Setting $\mathbf{Z}_1^* = 0$, we have $\mathrm{tr}((\mathbf{A}^* - \lambda\mathbf{I})\mathbf{Z}_2^*) = \mathrm{tr}((\mathbf{B} - \lambda\mathbf{I} - \mathbf{Z}_2^*)\mathbf{Z}_2^*) = 0$. Since $\mathbf{B}$ has at most $-s$ eigenvalues larger than λ, we have $\mathbf{Z}_\mathbf{2}^* = \max\{\lambda_1^\uparrow - \lambda, 0\}\mathbf{u}_1^\uparrow\mathbf{u}_1^{\uparrow\top} + \max\{\lambda_2^\uparrow - \lambda, 0\}\mathbf{u}_2^\uparrow\mathbf{u}_2^{\uparrow\top}$, where $(\mathbf{u}_1^\uparrow, \lambda_1^\uparrow)$ and $(\mathbf{u}_2^\uparrow, \lambda_2^\uparrow)$ correspond to the top two eigenpairs of the matrix $\mathbf{B}$ as defined in the Notation section.

Case 2: $\mathbf{s} \in \{\mathbf{1}, \mathbf{2}\}$ From Lemma 3, we have that all eigenvalues of $\mathbf{B}$ are not larger than λ. It is easy to verify that all eigenvalues of $\mathbf{A}^*$ are smaller than λ and hence $-(\mathbf{A}^* - \lambda\mathbf{I}) \in \mathbb{S}_+$. Similarly setting $\mathbf{Z}_2^* = 0$, we have $\mathrm{tr}(\mathbf{A}^*\mathbf{Z}_1^*) = \mathrm{tr}((\mathbf{B} + \mathbf{Z}_1^*)\mathbf{Z}_1^*) = 0$. Since $\mathbf{B}$ has at most s negative eigenvalues, we have $\mathbf{Z}_\mathbf{1}^* = -\min\{\lambda_1^\downarrow, 0\}\mathbf{u}_1^\downarrow\mathbf{u}_1^{\downarrow\top} - \min\{\lambda_2^\downarrow, 0\}\mathbf{u}_2^\downarrow\mathbf{u}_2^{\downarrow\top}$.

Case 3: $\mathbf{s} = 0$ From Lemma 3, we have that $\mathbf{B}$ at most one negative eigenvalue and also has at most one eigenvalue larger than λ. From Eqs. (9) we have

$$\begin{aligned} &\mathrm{tr}(\mathbf{A}^*\mathbf{Z}_1^*) = \mathrm{tr}((\mathbf{B} + \mathbf{Z}_1^* - \mathbf{Z}_2^*)\,\mathbf{Z}_1^*) = 0 \\ &\mathrm{tr}((\mathbf{A}^* - \lambda\mathbf{I})\mathbf{Z}_2^*) = \mathrm{tr}((\mathbf{B} - \lambda\mathbf{I} + \mathbf{Z}_1^* - \mathbf{Z}_2^*)\mathbf{Z}_2^*) = 0 \\ &\mathbf{Z}_1^* \in \mathbb{S}_+, \mathbf{Z}_2^* \in \mathbb{S}_+. \end{aligned} \tag{10}$$

Assuming the orthogonality between $\mathbf{Z}_1^*$ and $\mathbf{Z}_2^*$ and using similar analysis to Case 1 and Case 2, we have $\mathbf{Z}_\mathbf{1}^* = -\min\{\lambda_1^\downarrow, 0\}\mathbf{u}_1^\downarrow\mathbf{u}_1^{\downarrow\top}$ and $\mathbf{Z}_\mathbf{2}^* = \max\{\lambda_1^\uparrow - \lambda, 0\}\mathbf{u}_1^\uparrow\mathbf{u}_1^{\uparrow\top}$. It can be verified that $\mathbf{Z}_\mathbf{1}^*$ and $\mathbf{Z}_\mathbf{2}^*$ satisfy Eqs. (10) and they are optimal dual solutions.

The primal variable $\mathbf{A}^*$ can be computed from the first equation in Eqs. (9) using the obtained $\mathbf{Z}_1^*$ and $\mathbf{Z}_2^*$. This completes the proof. □

Algorithm 1 Rank-2-Thresholding $(\mathbf{B}, \lambda, s)$

if $s \in \{-1, -2\}$ **then**
  return $\mathbf{B} - [\lambda_1^\uparrow - \lambda]_+\mathbf{u}_1^\uparrow\mathbf{u}_1^{\uparrow\top} - [\lambda_2^\uparrow - \lambda]_+\mathbf{u}_2^\uparrow\mathbf{u}_2^{\uparrow\top}$
else if $s = 0$ **then**
  return $\mathbf{B} - [\lambda_1^\uparrow - \lambda]_+\mathbf{u}_1^\uparrow\mathbf{u}_1^{\uparrow\top} - [\lambda_1^\downarrow]_-\mathbf{u}_1^\downarrow\mathbf{u}_1^{\downarrow\top}$
else if $s \in \{1, 2\}$ **then**
  return $\mathbf{B} - [\lambda_1^\downarrow]_-\mathbf{u}_1^\downarrow\mathbf{u}_1^{\downarrow\top} - [\lambda_2^\downarrow]_-\mathbf{u}_2^\downarrow\mathbf{u}_2^{\downarrow\top}$
end if

We then present an efficient algorithm for solving Eq. (3) with the Frobenius norm constraint employed in the domain set $\mathcal{D}$. Note that the results in Theorem 5 can be proved using techniques similar to the ones in Theorem 5.

Theorem 5. *Let $\mathbf{A}$, $\mathbf{B}$, s_1, s_2, and γ be defined in Eq. (7) and denote $s = s_1 + s_2$. Denote $\widehat{\mathbf{B}} =$ Rank-2-Thresholding $(\mathbf{B}, +\infty, s)$ and $\tau = \|\widehat{\mathbf{B}}\|_F$. The optimal solution to*

$$\begin{aligned} \min_{\widehat{\mathbf{A}}} \quad & \frac{1}{2}\|\widehat{\mathbf{A}} - \mathbf{B}\|_F^2 \\ s.t. \quad & \widehat{\mathbf{A}} \in \mathbb{S}_+,\ \|\widehat{\mathbf{A}}\|_F \leq \lambda \end{aligned} \tag{11}$$

is given by $\mathbf{A}^* = \lambda\widehat{\mathbf{B}}/\max\{\lambda, \tau\}$.

5.2 Projection with Rank-k Stochastic Gradients

Similarly, considering the explicit expression of the rank-k stochastic gradient $\widehat{\mathcal{G}}_k(\mathbf{A}_t; \omega_t)$, we can rewrite $\mathbf{B}_t = \mathbf{A}_t - \gamma_t\widehat{\mathcal{G}}_k(\mathbf{A}_t; \omega_t)$ as

$$\mathbf{B} = \mathbf{A} - \gamma\sum_{i=1}^{k} s_i\mathbf{v}_i\mathbf{v}_i^\top,\ \mathbf{A} \in \mathcal{D}, \tag{12}$$

where $s_i \in \{-1, 0, +1\}$ and $\gamma > 0$.We present the main results of this subsection in the following theorem.

Algorithm 2 Rank-k-Thresholding $(\mathbf{B}, \lambda, s_+, s_-)$

return $\mathbf{B} - \sum_{j=1}^{s_+}[\lambda_j^\downarrow]_-\mathbf{u}_j^\downarrow\mathbf{u}_j^{\downarrow\top} - \sum_{i=1}^{s_-}[\lambda_i^\uparrow - \lambda]_+\mathbf{u}_i^\uparrow\mathbf{u}_i^{\uparrow\top}$

Algorithm 3 LR-SGD for Solving Eq. (1)

1: Initialize $\mathbf{A}_1 \in \mathcal{D}$.
2: **for** $t = 1, 2, \cdots$ **do**
3: Compute a low-rank stochastic gradient of $f(\cdot)$ at $\mathbf{A}_t$ as $\widehat{\mathcal{G}}_k(\mathbf{A}_t; \omega_t) = \sum_{i=1}^{k} s_i^t \mathbf{v}_i^t \mathbf{v}_i^{t\top}$
4: Update $\mathbf{A}_{t+1}$ via solving the SDP problem

$$\mathbf{A}_{t+1} = \arg\min_{\widehat{\mathbf{A}}} \quad \frac{1}{2}\|\widehat{\mathbf{A}} - (\mathbf{A}_t - \gamma_t \widehat{\mathcal{G}}_k(\mathbf{A}_t; \omega_t))\|_F^2$$
$$\text{s.t.} \quad \widehat{\mathbf{A}} \in \mathbb{S}_+, \ \|\widehat{\mathbf{A}}\| \le \lambda.$$

5: **end for**

Theorem 6. *Let $\mathbf{A}, \mathbf{B}, s_i$ and γ be defined in Eq. (12). For the sequence $\{s_i\}$, denote the number of positive entries and the number of negative entries by s_+ and s_-, respectively. Consider the SDP problem*

$$\min_{\widehat{\mathbf{A}}} \quad \tfrac{1}{2}\|\widehat{\mathbf{A}} - \mathbf{B}\|_F^2 \qquad (13)$$
$$s.t. \quad \widehat{\mathbf{A}} \in \mathbb{S}_+, \ \|\widehat{\mathbf{A}}\| \le \lambda,$$

where $\|\cdot\|$ is a matrix norm. Then

(1) if the spectral norm is used, i.e., $\|\widehat{\mathbf{A}}\|_2 \le \lambda$, the optimal solution to Eq. (13) is given by $\mathbf{A}^* =$ Rank-k-Thresholding $(\mathbf{B}, \lambda, s_+, s_-)$.

(2) if the Frobenius norm is used, i.e., $\|\widehat{\mathbf{A}}\|_F \le \lambda$, the optimal solution to Eq. (13) is given by $\mathbf{A}^* = \lambda\widetilde{\mathbf{A}}/\max\{\lambda, \tau\}$, where $\tau = \|\widetilde{\mathbf{A}}\|_F$ and $\widetilde{\mathbf{A}} =$ Rank-k-Thresholding $(\mathbf{B}, \lambda, s_+, s_-)$.

Similarly the results in Theorem 6 can be proved using techniques from Section 5.1.

6 Main Algorithm Example

The proposed low-rank stochastic gradients and their efficient projection algorithms can be combined with various stochastic based methods for solving the SDP problem in Eq. (1). For illustration, we use the standard stochastic (projected) gradient descent method as an example to demonstrate how such a combination, i.e., the low-rank stochastic gradient descent method (LR-SGD), efficiently solves Eq. (1).

The pseudo codes of LR-SGD are presented in Algorithm 3. Since LR-SGD is essentially a SGD method, it achieves the same convergence rate as the standard SGD, i.e., $\mathcal{O}(1/\sqrt{T})$ for general convex functions and $\mathcal{O}(\log T/T)$ for strongly convex functions, where T denotes the required iteration number [1, 33].

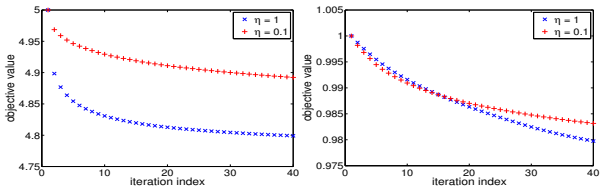

Figure 1: Convergence plots of LR-SGD for solving LMNN in Eq. (6): the left and right plots are obtained respectively by setting $c = 1$ and $c = 5$; c denotes the hinge loss parameter and $\eta/\sqrt{t}$ specifies the step size in the t-th iteration.

7 Experiments

We conduct numerical studies on LR-SGD in comparison with other representative algorithms to demonstrate its effectiveness and efficiency.

In the following experiments we use two real-world datasets, i.e., CiteSeer and Cora [34], as well as a synthetic data. CiteSeer includes 3312 scientific publications exclusively from 6 categories. Each publication is represented as a binary vector; the binary entries indicate the presence or absence of 3703 meaningful words; all vectors are normalized to unit length. Cora consists of 2708 scientific publications exclusively from 7 categories. Similarly each publication is denoted by a normalized binary vector of length 1433. All algorithms are implemented in Matlab and the simulation studies are conducted on an Intel Xeon 3.2GHZ CPU.

7.1 Convergence Study of LR-SGD

To study the practical convergence speed of LR-SGD, we use LR-SGD to solve LMNN in Eq. (6) and record the obtained intermediate objective values.

The experimental setup is as follows. From the CiteSeer data, we construct 6624 neighbor pairs (NP) by randomly selecting 2 neighbor publications (of the same class label) for each of the 3312 publications; we then construct 19872 non-neighbor triples (NNT) by randomly selecting 3 non-neighbor publications (of a different class label) for each NP. In each iteration of LR-SGD, we uniformly sample a NP (from the 6624 NPs) and a NNT (from the 19872 NNTs), respectively; using the sampled NP and NNT, we construct a low-rank stochastic gradient and update the optimization variable $\mathbf{A}$ towards the global optimum; we run the algorithm 20000 times and record the intermediate objective value every 500 iterations.

We use various values for the hinge loss parameter c and the step size parameter η in the experiments. For demonstration, we present 4 representative convergence plots in Figure 1. From the experimental results, we can observe: (1) the parameter η is critical for the practical convergence speed of LR-SGD; (2) setting $\eta = 1$ leads to faster practical convergence

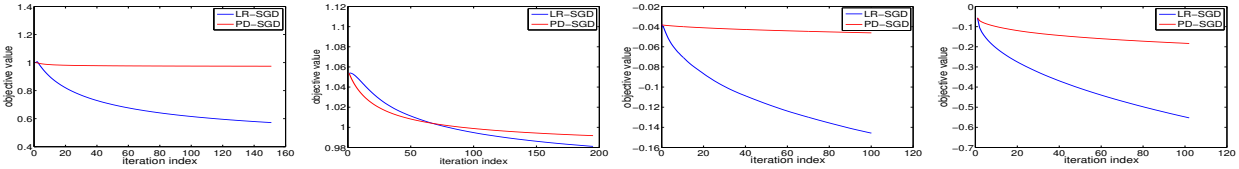

Figure 2: Comparison of LR-SGD and PD-SGD in term of their practical convergence speeds. The first two plots are obtained from solving LMNN with the constraint $\|\mathbf{A}\|_F \leq 1$ using CiteSeer and Cora, respectively. The second two plots are obtained from solving CML with the constraint $\|\mathbf{A}\|_1 \leq 1$ using CiteSeer and Cora, respectively. Note that each index on the x-axis represents 500 algorithmic iterations as we evaluate the objective value once every 500 iterations.

speed on this specific data, compared to the setting of using $\eta = 0.1$; (3) the demonstrated convergence speed is consistent with the theoretical analysis in Section 6, that is, LR-SGD convergence at the rate $\mathcal{O}(1/\sqrt{T})$ for general convex objective functions.

7.2 Comparison with a Competing Method

We compare LR-SGD and PD-SGD in terms of their practical convergence speeds. Specifically we employ LR-SGD and PD-SGD respectively for solving the LMNN formulation in Eq. (6) (with a unit Frobenius norm constraint $\|A\|_F \leq 1$) and the *Constrained Metric Learning formulation* (CML) [8, 9] (with a unit spectral norm constraint $\|\mathbf{A}\|_2 \leq 1$) as

$$\begin{aligned} \min_{\mathbf{A}} \quad & \frac{\gamma}{|\mathcal{S}|}\sum_{(i,j)\in\mathcal{S}} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 - \frac{1}{|\mathcal{N}|}\sum_{(i,j)\in\mathcal{N}} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 \\ \text{s.t} \quad & \|\mathbf{A}\|_2 \leq 1,\ \mathbf{A} \in \mathbb{S}_+, \end{aligned} \tag{14}$$

where $\mathcal{S}$ and $\mathcal{N}$ denote the set of similar pairs (SP) (of the same class labels) and the set of dissimilar pairs (DP) (of different class labels), respectively.

We construct data subsets from CiteSeer and Cora for the following experiments. Similar to Section 7.1, from CiteSeer we construct 6624 NPs and 331200 NNTs by randomly selecting 5 non-neighbors for each NP; we construct 6624 SPs and 331200 DPs by randomly selecting 5 samples of different labels for each SP. We apply a similar procedure on Cora to construct 5416 NPs and 5416 NNTs as well as 5416 SPs and 5416 DPs.

In each iteration of LR-SGD and PD-SGD, we randomly select a NP and a NNT (or a SP and a DP) with replacement to construct a low-rank stochastic gradient to update the optimization variable for solving LMNN (or for solving CML). We stop LR-SGD if the objective value change in two successive iterations is smaller than 10^{-3} or the maximum iteration number is larger than 10^6; we terminate PD-SGD when it runs the same number of iteration as LR-SGD. In our experiment, we set $c = 1$ for both LMNN and CML; the step size parameters are estimated from tuning.

The experimental results in Figure 2 demonstrate that LR-SGD attains smaller objective values than PD-SGD with an appropriate stopping criterion; in other words, LR-SGD has a faster practical convergence speed compared to PD-SGD. It is worth noting that PD-SGD has relatively less computation complexity per iteration, compared to LR-SGD. Our experimental results are consistent with this theoretical analysis; for example, for the first plot in Figure 2, the computation time for LR-SGD and PD-SGD are 9570 and 4311 minutes, respectively; for the third plot, the computation time for LR-SGD and PD-SGD are 1903 and 751 minutes, respectively. In our experiments LR-SGD attains smaller objective values than PD-SGD using the same amount of computation time; this result can also be inferred from the trend in Figures 2.

7.3 Study of the Low-Rank Approximation

We conduct two experiments to study the implicit construction scheme proposed in Section 4.2. Synthetic data is employed for the following experiments. The first experiment is used to verify the validity of the theoretical results in Lemmas 1 and 2. Specifically, we construct a symmetric (not necessarily PSD) matrix $\mathbf{G}$ of size 10×10 by sampling its entries from $\mathcal{N}(0, 1)$. Denote the diagonal and off-diagonal components of $\mathbf{G}$ respectively by $\mathbf{D}$ and $\mathbf{P}$. From Lemmas 1 and 2, we construct rank-one counterparts of $\mathbf{D}$ and $\mathbf{P}$, respectively, denoted by $\widehat{\mathbf{D}}$ and $\widehat{\mathbf{P}}$. We repeat the construction procedure 500 times and measure the difference of $\mathbf{D}$ and its averaged low-rank counterpart $\widehat{\mathbf{D}}$, and also measure the difference of $\mathbf{P}$ and its averaged low-rank counterpart $\widehat{\mathbf{P}}$.

We present the experimental results in Figure 3; in the first plot, we compare the diagonal entries from both $\mathbf{D}$ and the averaged $\widehat{\mathbf{D}}$, while in the second and the third plots, we display the off-diagonal matrix $\mathbf{P}$ and its averaged $\widehat{\mathbf{P}}$ with colormap set as gray. Clearly the experimental results empirically demonstrate $\mathbb{E}(\widehat{\mathbf{D}}) = \mathbf{D}$ and $\mathbb{E}(\widehat{\mathbf{P}}) = \mathbf{P}$, as well as verify the validity of the theoretical results in Lemmas 1 and 2.

The second experiment is used to study the relationship of the (low-rank) approximation error and the rank number. Specifically, we generate a symmetric matrix $\mathbf{G}$ of size 500×500 and then construct a low-rank matrix $\mathbf{G_k}$ of the same size following from

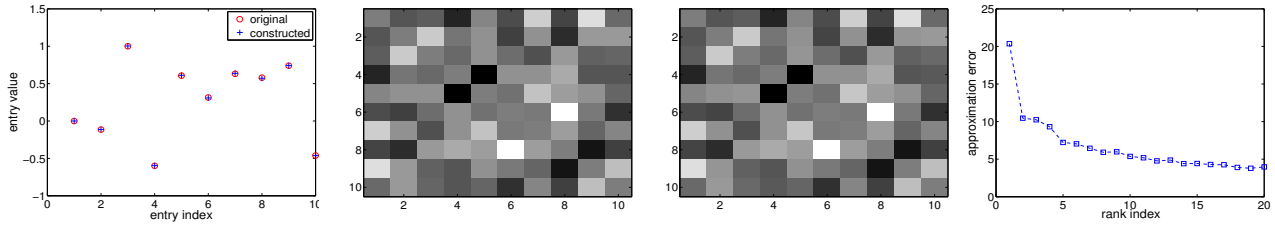

Figure 3: The first 3 plots show a comparison of the constructed diagonal matrix (from Lemma 1) and the off-diagonal matrix (from Lemma 2) respectively with their ground truth; in the first plot, the diagonal entries of D (original) are compared with the diagonal entries of the ground truth (constructed); the 2nd and the 3rd plots respectively represent the construction and ground truth of the off-diagonal matrix. The last plot shows low-rank approximation error using different ranks; the approximation error is defined as $\|\mathbf{G} - \widehat{\mathbf{G}}\|_F/d$.

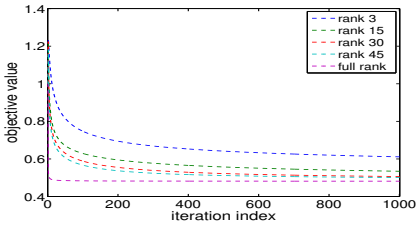

Figure 4: Convergence plots of LR-SGD using stochastic gradients of various rank values.

Lemmas 1 and 2. We vary the rank k in the range $20 \times \{1, 2, \cdots, 20\}$ and record the approximation error, i.e., $\|\mathbf{G} - \mathbf{G_k}\|_F/500$, for each value k.

The last plot of Figure 3 demonstrates that the approximation error decreases with the increase of the rank number (in the constructed matrix). This result is consistent with our hypothesis and also demonstrates the effectiveness of the low-rank approximation procedure.

7.4 Study on the Rank of the Stochastic Gradients

We study the effect of the stochastic gradient's rank values on the practical convergence speeds of LR-SGD. Specifically we use LR-SGD to solve a constrained version of the *Maximum-Margin Matrix Factorization* (MMMF) formulation [4] formulated as

$$\begin{aligned} \min_{\mathbf{A}} \quad & \lambda\|\mathbf{A}\|_1 + \textstyle\sum_{i,j\in\Omega} \ell\left(b - A_{ij}Y_{ij}, 0\right) \\ \text{s.t} \quad & \|\mathbf{A}\|_F \leq 1,\ \mathbf{A} \in \mathbb{S}_+, \end{aligned} \tag{15}$$

where Y_{ij} is equal to 1 if sample i and sample j have the same class label and -1 otherwise, Ω denotes the pairwise constraint set, and $\ell(\cdot)$ denotes the hinge loss.

The experimental setup is as follows. We generate 100 samples and evenly assign them to two classes. To construct the set Ω, we uniformly sample 200 sample pairs: half of them have the same class label and the other half have different class labels. For illustration, we set $\lambda = 1$ and $b = 0.5$ in Eq. (15); for other parameter settings, we observe similar trends. We compute a stochastic gradient for the objective function of Eq. (15) as follows: from Lemma 1, sample a rank-k stochastic gradient from the subgradient (an identity matrix) of the trace norm term; from Lemmas 1 and 2, sample a rank-$2k$ stochastic gradient from a subgradient of the hinge loss term; compute the sum of the obtained two stochastic gradients. This procedure leads to a stochastic gradient of rank no larger than $3k$.

Table 1: Computation time in seconds (Tim.) for computing various numbers of eigenpairs (Num.) from a symmetric matrix of size $10^5 \times 10^5$.

Num.	2	50	500	1000	2500	5000
Tim.	1.1	8.6	458.5	1454.9	8318.8	30435.6

We vary the rank value of the stochastic gradients in LR-SGD and record the intermediate objective values obtained using such a stochastic gradient. The experimental results in Figure 4 consistently show that stochastic gradients of higher ranks lead to faster convergence speed; this is due to the fact that more information is embedded in the stochastic gradients of higher ranks. However, computing a stochastic gradient of a higher rank value obviously needs more computation time, i.e., computing more eigenpairs require more computation time, as depicted in Table 1. There is a trade-off between the practical convergence speed and the required computation time.

8 Conclusions

We presented LR-SGD for solving a class of SDP problems. By analyzing the low-rank structure of the stochastic gradients, LR-SGD efficiently solves the SDP projection involved in each of its iterations, demonstrating clear computational advantages. Interestingly, our theoretical analysis reveals the universal existence of low-rank stochastic gradients for a class of SDP problems and consequently validates the rationale of LR-SGD. We have applied LR-SGD to solving distance metric learning problems in comparison with a representative SGD method. The experimental results demonstrate the effectiveness and efficiency of the proposed algorithms. In the future, we plan to conduct theoretical studies to estimate appropriate rank values for the stochastic gradients; we also plan to apply LR-SGD to solving more general SDP problems with real-world datasets.

References

[1] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM J. on Optimization*, vol. 19, pp. 1574–1609, 2009.

[2] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, 2009.

[3] A. d'Aspremont, O. Banerjee, and L. El Ghaoui, "First-order methods for sparse covariance selection," *SIAM J. Matrix Anal. Appl.*, vol. 30, pp. 56–66, 2008.

[4] S. Nathan, D. M. R. Jason, and S. J. Tommi, "Maximum margin matrix factorization," in *NIPS*, 2005.

[5] Y. Zhang and Y. Dit-Yan, "A convex formulation for learning task relationships in multi-task learning," in *UAI*, 2010.

[6] J. Chen, L. Tang, J. Liu, and J. Ye, "A convex formulation for learning a shared predictive structure from multiple tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, pp. 1025–1038, 2013.

[7] A. d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. G. Lanckriet, "A direct formulation for sparse pca using semidefinite programming," *SIAM Rev.*, vol. 49, pp. 434–448, 2007.

[8] W. Liu, X. Tian, D. Tao, and J. Liu, "Constrained metric learning via distance gap maximization." in *AAAI*, 2010.

[9] S. C. Hoi, W. Liu, and S.-F. Chang, "Semi-supervised distance metric learning for collaborative image retrieval and clustering," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 6, pp. 18:1–18:26, 2010.

[10] M. Mahdavi, T. Yang, R. Jin, S. Zhu, and J. Yi, "Stochastic gradient descent with only one projection," in *NIPS*, 2012.

[11] E. Hazan and S. Kale, "Stochastic gradient descent with only one projection," in *ICML*, 2012.

[12] M. Frank and P. Wolfe, "An algorithm for quadratic programming. naval research logistics," *Naval Res. Logistics*, vol. 3, pp. 95–110, 1956.

[13] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Netherlands, 2003.

[14] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *ICML*, 2007.

[15] S. Shalev-Shwartz, Y. Singer, and A. Y. Ng, "Online and batch learning of pseudo-metrics," in *ICML*, 2004.

[16] P. Jain, B. Kulis, I. S. Dhillon, and K. Grauman, "Online metric learning and fast similarity search," in *NIPS*, 2008, pp. 761–768.

[17] H. Avron, S. Kale, S. Kasiviswanathan, and V. Sindhwani, "Efficient and practical stochastic subgradient descent for nuclear norm regularization," in *ICML*, 2012.

[18] S. Arora, E. Hazan, and S. Kale, "Fast algorithms for approximate semide.nite programming using the multiplicative weights update method," in *FOCS*, 2005.

[19] E. Hazan, "Sparse approximate solutions to semidefinite programs," in *LATIN*, 2008.

[20] A. Kleiner, A. Rahimi, and M. I. Jordan, "Random conic pursuit for semidefinite programming," in *NIPS*, 2010.

[21] S. Laue, "A hybrid algorithm for convex semidefinite optimization," in *ICML*, 2012.

[22] D. Garber and E. Hazan, "Approximating semidefinite programs in sublinear time," in *NIPS*, 2011.

[23] Y. Nesterov, "Smoothing technique and its applications in semidefinite optimization," *Math. Program.*, pp. 245–259, 2007.

[24] Y. Ying and P. Li, "Distance metric learning with eigenvalue optimization," *J. Mach. Learn. Res.*, pp. 1–26, 2012.

[25] K. Weinberger and L. Saul, "Fast solvers and efficient implementations for distance metric learning," in *ICML*, 2008.

[26] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre, "Generalized power method for sparse principal component analysis," *J. Mach. Learn. Res.*, vol. 11, pp. 517–553, 2010.

[27] A. d'Aspremont, F. Bach, and L. E. Ghaoui, "Optimal solutions for sparse principal component analysis," *J. Mach. Learn. Res.*, vol. 9, pp. 1269–1294, 2008.

[28] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[29] Z. Wen, D. Goldfarb, and W. Yin, "Alternating direction augmented lagrangian methods for semidefinite programming," *Mathematical Programming Computation*, 2010.

[30] H. Ouyang, N. He, L. Tran, and A. G. Gray, "Stochastic alternating direction method of multipliers," in *ICML*, 2013.

[31] N. Srebro and A. Tewari, "Stochastic optimization for machine learning," in *ICML Tutorial*, 2010.

[32] J. N. Franklin, *Matrix Theory*. Courier Dover Publications, 2000.

[33] J. Duchi, S. Shalev-Shwartz, Y. Singer, and A. Tewari, "Composite objective mirror descent." in *COLT*, 2010.

[34] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, 2008.