

Sample Average Approximation for Stochastic Optimization with Dependent Data: Performance Guarantees and Tractability

Yafei Wang,¹ Bo Pan,¹ Wei Tu,² Peng Liu,³ Bei Jiang,¹ Chao Gao,⁴ Wei Lu,⁵
Shangling Jui,⁶ Linglong Kong^{1*}

¹University of Alberta, ²Queen’s University, ³University of Kent, ⁴Huawei Canada Research Center,
⁵Huawei Canada, ⁶Huawei Technologies Ltd.
{yafei2, pan1, bei1, lkong}@ualberta.ca, wei.tu@queensu.ca, p.liu@kent.ac.uk, chao.gao4@huawei.com,
robin.luwei@hisilicon.com, jui.shangling@huawei.com

Abstract

Sample average approximation (SAA), a popular method for tractably solving stochastic optimization problems, enjoys strong asymptotic performance guarantees in settings with independent training samples. However, these guarantees are not known to hold generally with dependent samples, such as in online learning with time series data or distributed computing with Markovian training samples. In this paper, we show that SAA remains tractable when the distribution of unknown parameters is only observable through dependent instances and still enjoys asymptotic consistency and finite sample guarantees. Specifically, we provide a rigorous probability error analysis to derive $1 - \beta$ confidence bounds for the out-of-sample performance of SAA estimators and show that these estimators are asymptotically consistent. We then, using monotone operator theory, study the performance of a class of stochastic first-order algorithms trained on a dependent source of data. We show that approximation error for these algorithms is bounded and concentrates around zero, and establish deviation bounds for iterates when the underlying stochastic process is ϕ -mixing. The algorithms presented can be used to handle numerically inconvenient loss functions such as the sum of a smooth and non-smooth function or of non-smooth functions with constraints. To illustrate the usefulness of our results, we present several stochastic versions of popular algorithms such as stochastic proximal gradient descent (S-PGD), stochastic relaxed Peaceman–Rachford splitting algorithms (S-rPRS), and numerical experiment.

Introduction

Stochastic optimization, a powerful modeling paradigm in optimization under uncertainty, is ubiquitous in statistical machine learning, engineering, and decision-making problems (Franklin 2005; Heyman and Sobel 2004; Fouskakis and Draper 2002). Specifically, these problems seek to minimize an expected loss taken with respect to the distribution \mathbb{P} of a random parameter ξ . However, more often than not, this probability distribution is unknown and can only be observed through a finite number of sample points. We are thus forced to solve a surrogate optimization problem constructed by the observed data: the optimal value of the

original problem can be approximated by that of the surrogate problem. The main goal of this paper is to study the properties of sample average approximation (SAA), a powerful approach to stochastic optimization that is considered statistically and computationally “optimal” in settings where observations are independent. These claims have the following caveats in non-independent settings.

- Most existing statistical guarantees for SAA critically depend on the assumption that training samples are independent and identically distributed (i.i.d.). However, the i.i.d. assumption can be difficult to justify or outright invalid in practice. It is thus important to examine the properties of SAA estimators when samples are known to be correlated.
- For an optimization scheme to be useful, it should be solved efficiently. Standard convex optimization techniques, while widely applicable, suffer performance-wise in problems that are complex or highly structured, and trained with non-i.i.d. samples. Consequently, practically useful methods should offer guarantees that remain valid when the training samples display serial dependencies and be rich enough to handle numerically inconvenient problems.

In this paper, we consider applying SAA to the stochastic optimization problem

$$J^* = \min_{x \in \mathbb{X}} \left\{ \mathbb{E}^{\mathbb{P}}[\ell(x; \xi)] = \int_{\Xi} \ell(x; \xi) \mathbb{P}(d\xi) \right\}, \quad (1)$$

where Ξ is a sample space with a probability distribution \mathbb{P} and $\mathbb{X} \in \mathbb{R}^d$ is a convex, feasible parameter space. We assume throughout that $\ell(x; \xi)$ is a closed, convex, and proper function and that $\xi \in \Xi$ denotes a sample instance. We let x^* denote the optimal solution to problem (1).

In most situations of practical interest, the distribution \mathbb{P} is not known or cannot be efficiently sampled, such as when Ξ is a high-dimensional or combinatorial sample space (Johansson, Rabi, and Johansson 2007, 2010). This restriction removes information essential to solving problem (1) exactly. We instead consider receiving samples $\{\xi_k\}_{k=1}^K$ from a stochastic process $P = P^k$ indexed by time k , where P converges to the stationary distribution \mathbb{P} . This is a natural relaxation of the assumption that training samples are i.i.d. following \mathbb{P} . As an example, consider $\Xi =$

*Corresponding author

$\{\xi \in \{0, 1\}^d \mid \langle a, \xi \rangle \leq b\}$, where $a \in \mathbb{R}^d, b \in \mathbb{R}$, $\langle a, \xi \rangle = \sum_{i=1}^d a_i \xi_i$, and \mathbb{P} is the uniform distribution over Ξ . A straightforward way to obtain a sample from \mathbb{P} is by iterative random sampling from $\{0, 1\}^d$ until the constraint on Ξ is satisfied: this approach takes $O(2^d)$ draws to obtain a feasible sample. Alternatively, it is possible to design a Markov chain (Jerrum and Sinclair 1996) that generates a sample that is ε -close to the distribution \mathbb{P} and only requires $\log(\sqrt{d}/\varepsilon) \exp(O(\sqrt{d}(\log d)^{5/2}))$ draws (Dyer et al. 1993), a greatly reduced sampling cost. Autoregressive processes (Kushner and Yin 2003) are yet another example of stochastic data-generating processes but generate a dependent source of data, here the sequential entries of a time series. The assumption of i.i.d. samples is, therefore, unrealistic in many data-generating processes. These two examples highlight the need to consider sampling efficiency and training under inter-sample dependence.

Applications of SAA to stochastic optimization are not new and have been studied extensively in the literature (Kleywegt, Shapiro, and Homem-de Mello 2002; Kim, Pasupathy, and Henderson 2015; Emelogu et al. 2016; Bertsimas, Gupta, and Kallus 2018). As noted above, the idea underlying SAA is simple—to generate solutions to problem (1), approximate \mathbb{P} with the discrete empirical distribution $\hat{\mathbb{P}}_K = \frac{1}{K} \sum_{k=1}^K \delta_{\xi_k}$ corresponding to training samples. SAA improves problem solvability by turning integration over a density function in a summation over discrete points.

Properties of solutions to SAA problems are well understood. In particular, the optimal solution of an SAA problem is known to be strongly consistent and asymptotically normal (Kim, Pasupathy, and Henderson 2015). However, most works study problems in settings where data is i.i.d. With the notable exceptions of Duchi et al. (2012) and Agarwal and Duchi (2012), we are not aware of any studies of SAA with non-i.i.d. data, while these work focus more on the convergence of the proposed algorithm and iterate asymptotics rather than the properties of SAA. Results on the asymptotic consistency of SAA under an unknown probability distribution \mathbb{P} and dependent training data have not yet been established and are of particular importance.

Tractability is equally as important as statistical guarantees when establishing the practical utility of an optimization scheme. SAA tractability suffers when the loss function ℓ possesses a complex structure, such as statistical machine learning problems that enforce prior knowledge of the form of the solution, such as sparsity, low rank, and smoothness (Franklin 2005). As a result, it is critically important to develop algorithms that are both rich enough to capture the complexity of data and scalable enough to process data in a parallelized or fully decentralized fashion. We study the tractability of SAA with non-i.i.d. training samples for a class of stochastic first-order algorithms. We focus primarily on operator-splitting schemes, which are widely used due to their scalability with respect to problem dimensionality. More importantly, operator splitting schemes can be easily parallelized and are usually simple and cheap to implement.

One general iteration scheme, defined as the Stochastic

Krasnosel’skiĭ–Mann (S-KM) iteration, is

$$x^k = x^{k-1} + \lambda_{k-1}(T(x^{k-1}) - x^{k-1} + \epsilon_{k-1})$$

where T is a nonexpansive operator defined as a mapping: $T : \mathbb{X} \rightarrow \mathbb{X}$ such that $\|Tx - Ty\| \leq \|x - y\|$ holds for all $x, y \in \mathbb{X}$. The stochastic error ϵ_{k-1} is caused by uncertainty in random sampling. Succinctly, an operator splitting algorithm converts an optimization problem into a problem of finding a fixed point of a nonexpansive operator and breaks this problem into several relatively simple subproblems. Many commonly used methods, such as stochastic proximal gradient descent (PGD) and stochastic alternating direction method of multipliers (ADMM), have this iteration step with a specific nonexpansive operator. Classical PGD and ADMM algorithms are special cases of the KM iteration without the stochastic error term.

Our setting is perhaps most similar to that in Duchi et al. (2012), which also considers receiving data from an ergodic process. However, our work here differs in two fundamental ways. First, Duchi et al. (2012) focuses on algorithmic convergence guarantees with dependent samples but pays little attention to statistical properties of SAA estimators. Second, Duchi et al. (2012) only considers stochastic mirror descent while we consider multiple other algorithms. Sun, Sun, and Yin (2018) similarly only considers stochastic gradient descent and works under the assumption that training samples are generated by a Markov chain, a special case of an ergodic process. The sampling technique used in Derman and Mannor (2020) is the same as that in Sun, Sun, and Yin (2018), except that samples are generated from multiple independent trajectories of serially correlated states. Using multiple replication (MR), a technique that attempts to remove inter-sample dependence via multiple stochastic process trajectories, the authors generate i.i.d. samples and train the model using standard algorithm. This approach can help to get rid of dependence among the samples but may be difficult to acquire in practice (we refer to the next section for details). In contrast to this approach, we establish asymptotic consistency and non-asymptotic bounds for SAA problems and study, from a fixed-point iteration perspective, properties of a variety of first-order algorithms based on a *single* trajectory of a stochastic process.

In summary, while many existing works indicate that SAA estimators are asymptotically consistent and tractable, hardly any existing statistical performance guarantees apply when training data fails to be i.i.d. or when training samples are generated from a single trajectory. This paper addresses this gap. Specifically, we study the performance of SAA estimators when training samples are generated by an ergodic stochastic process and, in the same setting, establish properties of the S-KM iteration in solving SAA. The main contributions of this paper are as follows.

- **Asymptotic consistency.** We generalize SAA problems to scenarios where training samples are correlated and prove that the SAA solutions are asymptotically consistent.
- **Finite sample guarantees.** By introducing a weakened version of ϕ -mixing, we establish $1 - \beta$ confidence

bounds on the out-of-sample performance based on the optimal solution obtained by minimizing an SAA problem.

- **Tractability.** We examine the performance of first-order algorithms for solving SAA problems in an efficient manner via monotone operator theory. We show that the approximation error of the algorithm is bounded and concentrates around zero, and further establish iterate deviation bounds.

SAA with Dependent Data

To motivate the broad applicability of sampling from a stochastic process, in this section, we begin with an example in distributed optimization under a simple peer-to-peer communication scheme (Johansson, Rabi, and Johansson 2007), where the optimization problem evolves according to a finite-state Markov chain. We then move toward a specific sampling technique, called the multiple replication approach, which generates training samples from multiple trajectories to get rid of dependency among the samples. However, its efficiency can not be guaranteed and it wastes a lot of samples. We then propose an iteration procedure that efficiently uses every obtained sample and works when there is only a single trajectory available.

Peer-to-Peer Optimization Suppose that the distribution \mathbb{P} is supported on a set of n points $\{\xi_1, \dots, \xi_n\}$ and that there are n processors, each with a convex function $\ell(x; \xi_i)$. The objective is to minimize $L(x) = \frac{1}{n} \sum_{i=1}^n \ell(x; \xi_i)$. To solve this problem, the current set of parameters $x^k \in \mathbb{X}$ is passed to one of the processors and updated in each iteration. More specifically, let the token $i(k)$ indicate the processor i holds x^k at time k : at this time, only the data stored in processor i is accessed. Given the current state $i(k)$, the next state $i(k+1)$ is determined randomly via $\mathbb{P}(i(k+1) = j \mid i(k) = i) = P_{ij}$, with $0 \leq P_{ij} \leq 1$. The data stored in processor j is then accessed and used to update x^k . Since Ξ is a combinational space in this setting, it is hard to draw samples directly. Moreover, it is unrealistic to assume that samples are i.i.d. However, because the token $i(k)$ can be viewed as evolving according to a Markov chain with a doubly stochastic transition matrix $(P_{ij})_{i,j}$, the data generating process forms a Markov chain.

Multiple Replication Approach As mentioned previously, although we can design a stochastic process to generate training samples, it may be impossible to generate i.i.d. training samples. A natural method called multiple replication approach (Gelman and Rubin 1992), is adopted to obtain a sequence of i.i.d. samples. Specifically, in this approach, after specifying the initial conditions of the stochastic process P , a sequence ξ_1, \dots, ξ_s , for some s , is generated. And we only keep the last sample ξ_s that follows the marginal distribution P^s , which is assumed to be close to \mathbb{P} . The same procedure is repeated for K times to simulate K i.i.d. samples, then use standard algorithms as for independent data.

Unfortunately, this method does not work if there is only one trajectory or expensive to simulate multiple trajec-

ries. Further, the computation cost could be quite high. This is because sampling a long trajectory and using only the last sample wastes a large number of samples, especially when s is large. This waste may seem necessary because a small s induces a large bias in ξ_s : after all, a random trajectory may take a long time to explore the parameter space and will often double back to previously visited states. This further complicates the problem of choosing a s appropriately. A small s will cause large bias in ξ_s , which slows the convergence of algorithms and reduces its final accuracy— $\{\xi_k\}_{k=1}^K$ are generated from P^s where P^s could be far away from \mathbb{P} . A large s , on the other hand, is wasteful especially when the iterate x^k is still far from convergence and some bias does not prevent the iteration update to make good progress. Therefore, s should increase adaptively as k increases—this makes the choice of s even more difficult.

Stochastic Krasnosel’skiĭ–Mann (S-KM) Iteration Assume that \mathbb{P} can only be observed through samples $\{\xi_k\}_{k=1}^K$ from an ergodic stochastic process P that converges to \mathbb{P} . We address the SAA problem

$$\hat{J}_K^* = \min_{x \in \mathbb{X}} \left\{ L(x) = \mathbb{E}^{\hat{\mathbb{P}}^K} [\ell(x; \xi)] = \frac{1}{K} \sum_{k=1}^K \ell(x; \xi_k) \right\} \quad (2)$$

and its corresponding optimal solution \hat{x}_K^* with an alternative iteration procedure that uses every sample immediately. Specifically, suppose that the distribution \mathbb{P} is supported on $\{\xi_k\}_{k=1}^K$. Problem (1) can then be approximated by (2). Iteration procedure to solve problem (2) is given in Algorithm 1: in the t -th iteration, the update

$$x^k \leftarrow x^{k-1} + \lambda_{k-1} (T(x^{k-1}; \xi_k) - x^{k-1})$$

is applied, where ξ_k is sampled from the stochastic process P evaluated at time k . The operator T in Algorithm 1 is a nonexpansive operator that depends on the specific method used. We include the sample ξ_k as an argument of T to explicitly indicate that the k -th iteration depends only on the most recently drawn sample. For more details regarding the forms of ℓ and T used in practice, please see the section of Application and Table 1.

Algorithm 1: Stochastic Krasnosel’skiĭ–Mann (S-KM)

Input: Initial value x^0 and given δ -optimality

While $\|T(\bar{x}^{k-1}; \xi^k) - \bar{x}^{k-1}\|^2 > \delta$

1: Sample $\xi^k \sim P^k$

2: $x^k \leftarrow \bar{x}^{k-1} + \lambda_{k-1} (T(\bar{x}^{k-1}; \xi^k) - \bar{x}^{k-1})$

3: $\bar{x}^k \leftarrow \frac{k-1}{k} \bar{x}^{k-1} + \frac{1}{k} x^k$

end while

Statistical Guarantees

In this section, we show that the widely used SAA method retains its statistical guarantees if training samples are generated by an ergodic stochastic process that converges to a desired stationary distribution \mathbb{P} . The reason for this is that the empirical distribution is a sufficient statistic and satisfies

requirements for large number theory. We begin with finite sample performance. If we only have access to K training samples, we can obtain the optimal solution \hat{x}_K^* and the corresponding optimal value \hat{J}_K^* via (2). The quality of \hat{x}_K^* and \hat{J}_K^* can be evaluated through out-of sample performance, defined as $\mathbb{E}^{\mathbb{P}}[\ell(\hat{x}_K^*; \xi_{\text{test}})]$, where ξ_{test} is a testing sample assumed to be drawn from \mathbb{P} and is independent of the training samples.

Preliminaries

We start the section by recalling some definitions that facilitate to present theoretical properties in the next section. The definition of total variation distance is first introduced to measure the convergence of the stochastic process P to the distribution \mathbb{P} .

Definition 1 Let \mathbb{P} and \mathbb{Q} be probability measures defined on a set Ξ with respective densities p and q relative to an underlying measure μ . The total variation distance between \mathbb{P} and \mathbb{Q} is

$$d_{\text{TV}}(\mathbb{P}, \mathbb{Q}) = \frac{1}{2} \int_{\Xi} |p(\xi) - q(\xi)| d\mu(\xi) = \sup_{A \subseteq \Xi} |\mathbb{P}(A) - \mathbb{Q}(A)|,$$

where the supremum is taken over measurable subsets of Ξ .

Using total variation distance, we can define the notion of a mixing stochastic process. Let $P_{[s]}^k = P^k(\cdot | \mathcal{F}_s)$ denote the distribution of ξ_k conditional on the σ -algebra \mathcal{F}_s with $\mathcal{F}_s = \sigma(\xi_1, \dots, \xi_s)$.

Definition 2 Define $\mathcal{F}_0 = \{\emptyset, \Omega\}$ and let $(\mathcal{F}_k)_{k=1}^K$ be an increasing sequence of σ -algebras such that $\mathcal{F}_{k-1} \subseteq \mathcal{F}_k$ for any k . The ϕ -mixing coefficient of the sample distribution P under total variation is

$$\phi(l) = \sup_{k \in \mathbb{N}^+, B \in \mathcal{F}_k} \{2d_{\text{TV}}(P^{k+l}(\cdot | B), \mathbb{P})\}.$$

We say that the process is ϕ -mixing if $\phi(l) \rightarrow 0$ as $l \rightarrow \infty$. Note that if the training samples are i.i.d., then $\phi(1) = 0$. We state the following results in a general form using ϕ -mixing coefficients.

Assumption and Main results

Before formalizing any statistical properties, we introduce one assumption.

Assumption 1 The ϕ -mixing coefficients for the sample distribution are summable, i.e., $\sum_{k=1}^{\infty} \phi(k) < \infty$.

Assumption 1 is met by some stochastic processes satisfying geometric mixing since $\phi(k) \leq \phi_0 \exp(-\phi_1 k^\alpha)$ would hold for some $\phi_0 > 0$, $\phi_1 > 0$, and $\alpha > 0$. A large class of stochastic processes are geometric mixing: this includes autoregressive models and aperiodic Harris-recurrent Markov processes (Modha and Masry 1996).

Let $\|\hat{\mathbb{P}}_K - \mathbb{P}\| = \int_0^1 |\hat{\mathbb{P}}_K(t) - \mathbb{P}(t)| dt$ with $\hat{\mathbb{P}}_K = \frac{1}{K} \sum_{k=1}^K \delta_{\xi_k}$. If Assumption 1 holds, then Theorem 1 in Dedecker and Merlevede (2007) indicates that

$$\mathbb{P} \left\{ \|\hat{\mathbb{P}}_K - \mathbb{P}\| \geq \varepsilon \right\} \leq 2 \exp \left(-\frac{K^2 \varepsilon^2}{2C(\sum_{k=1}^K \phi(k))} \right)$$

for all $K \geq 1$ and $\varepsilon > 0$, where $C(\sum_{k=1}^K \phi(k))$ is a function of $\sum_{k=1}^K \phi(k)$ and satisfies $C(\sum_{k=1}^K \phi(k)) < \infty$. This concentration inequality provides a prior estimate of the distribution \mathbb{P} that resides outside of the ε -ball $\mathbb{B}_\varepsilon(\hat{\mathbb{P}}_K) = \{\tilde{\mathbb{P}} | \|\hat{\mathbb{P}}_K - \tilde{\mathbb{P}}\| \leq \varepsilon\}$. Therefore, taking ε as

$$\varepsilon_K(\beta) = \left(\frac{2C(\sum_{k=1}^K \phi(k)) \log(2\beta^{-1})}{K^2} \right)^{\frac{1}{2}}, \quad (3)$$

we get the smallest ball that contains \mathbb{P} with confidence $1 - \beta$ for some prescribed $\beta \in (0, 1)$.

Theorem 1 (*Out-of-sample guarantees*) Let \hat{J}_K^* and \hat{x}_K^* be as defined in (2). Suppose that $\ell(x; \xi)$ is bounded by a constant L for $x \in \mathbb{X}$ and $\xi \in \Xi$. Let $\varepsilon^* = L\varepsilon_K(\beta)$. Then, we have that

$$\mathbb{P} \left\{ \mathbb{E}^{\mathbb{P}}[\ell(\hat{x}_K^*; \xi_{\text{test}})] \leq \hat{J}_K^* + \varepsilon^* \right\} \geq 1 - \beta.$$

Equation (3) indicates that $\varepsilon^* \rightarrow 0$ as $K \rightarrow \infty$ for any fixed β . Since the true distribution \mathbb{P} is unknown, the out-of-sample performance of \hat{x}_K^* , defined as $\mathbb{E}^{\mathbb{P}}[\ell(\hat{x}_K^*; \xi)]$, cannot be evaluated in practice. It is then more practical to establish bounds on $\mathbb{E}^{\mathbb{P}}[\ell(\hat{x}_K^*; \xi)]$. It can be seen directly that $J^* \leq \mathbb{E}^{\mathbb{P}}[\ell(\hat{x}_K^*; \xi)]$, but this lower bound is still impractical unless \mathbb{P} is known. Our primary concern here is to bound the cost from above. From Theorem 1, we can conclude that the out-of-sample performance of \hat{x}_K^* is bounded by a ball of \hat{J}_K^* with radius ε^* with probability $1 - \beta$. Esfahani and Kuhn (2018) establishes a similar results for minimization-maximization problems in the i.i.d. setting. In addition, one can show that if β_K converges to zero at a particular rate, then the solution to problem (2) converges to the original solution of problem (1) as K tends to infinity.

Theorem 2 (*Asymptotic consistency*) Let $\beta_K \in (0, 1)$ with $\lim_{K \rightarrow \infty} \varepsilon_K(\beta_K) = 0$ and $\sum_{K=1}^{\infty} \beta_K < \infty$. Under the assumptions of Theorem 1, we have

$$\mathbb{P} \left\{ \lim_{K \rightarrow \infty} \hat{J}_K^* = J^* \right\} = 1 \quad \text{and} \quad \mathbb{P} \left\{ \lim_{K \rightarrow \infty} \hat{x}_K^* = x^* \right\} = 1.$$

Computational Tractability

Even though SAA offers powerful statistical guarantees, it is practically useless unless the underlying optimization problem can be solved efficiently. In this section, we develop a numerical procedure to solve problem (2) when the data comes from an ergodic process P that converges to \mathbb{P} . We consider two types of problems related to problem (2): one unconstrained problem, given by

$$\min_{x \in \mathbb{X}} \left\{ \mathbb{E}^{\hat{\mathbb{P}}_K}[\ell(x; \xi)] = \frac{1}{K} \sum_{k=1}^K f(x; \xi_k) + g(x; \xi_k) \right\}, \quad (4)$$

and another subject to a linear constraint,

$$\min_{x \in \mathbb{X}, y \in \mathbb{X}} \left\{ \mathbb{E}^{\hat{\mathbb{P}}_K}[\ell(x, y; \xi)] = \frac{1}{K} \sum_{k=1}^K f(x; \xi_k) + g(y; \xi_k) \right\}$$

subject to $Ax + By = b$, (5)

where $b \in \mathbb{X}$ and the operators A, B are bounded and linear.

Many optimization problems can be cast as one of problem (4) or (5) (Zhang 2004; Teo et al. 2010; Davis and Drusvyatskiy 2019; Yu et al. 2019). Problems of these types arise in diverse applications in image processing, machine learning, and statistics (Boyd and Vandenberghe 2004; Pietrosanu et al. 2020, 2021; Wang et al. 2019; Zhang et al. 2021). In these fields, the dimensionality of data can be extremely large. Traditional methods may thus fail to efficiently (in terms of time) generate solutions. Regularizers, for example, enforce prior knowledge of the form of the solution, such as sparsity, low rank, or smoothness. In regularization schemes, f and g can be data fitting and penalty terms, respectively. Typically, penalty terms make problems (4) and (5) difficult to optimize jointly. Even if the both terms can be handled jointly, modern data is often high-dimensional and consists of millions or billions of training examples: running even a single iteration using classical algorithms is often infeasible. Moreover, in most statistical learning problems, we are more concerned with target parameter estimates rather than the objective function value. We present a stochastic KM algorithm that can handle a amount of non-i.i.d. data in a fast, parallelized, and efficient manner.

The methodology we present is different from those used in classical convex analysis (Boyd and Vandenberghe 2004), mainly because operator splitting algorithms are driven by fixed-point iteration rather than by the goal to minimize a loss function: convergence is due to the contraction property of a given fixed-point operator instead of “descent” on the loss. Most fixed-point iteration schemes do not decrease the objective function monotonically. Therefore, convergence of the objective function is a consequence of fixed-point convergence but not the cause of it.

For a nonexpansive operator $T : \mathbb{X} \rightarrow \mathbb{X}$, define $\text{Fix}(T) = \{x \in \mathbb{X} : x = T(x)\}$. We assume that $\text{Fix}(T) \neq \emptyset$. Let $\lambda_k \in (0, 1)$ and choose x^0 arbitrarily from \mathbb{X} . Then the S-KM iteration of T with data generated from P at time k is

$$\begin{aligned} x^k &= x^{k-1} + \lambda_{k-1} (T(x^{k-1}) - x^{k-1} + \epsilon_{k-1}) \\ &= T_{\lambda_{k-1}}(x^{k-1}) + \lambda_{k-1} \epsilon_{k-1}, \end{aligned}$$

where ϵ_{k-1} is caused by the randomness of samples.

The convergence of the fixed point iteration with a nonexpansive operator T fails in general. The S-KM algorithm thus replaces T with an averaged version T_λ to ensure convergence, as an averaged nonexpansive operator has the contraction property (Davis and Yin 2016). It can be shown that the fixed point of a nonexpansive operator is also the fixed point of its corresponded averaged nonexpansive operator. In practice, the operator T in Algorithm 1 depends on the stochastic splitting method used. For example, when using stochastic proximal gradient algorithm to solve problem (4), $T = \mathcal{J}_{\gamma \partial g} \circ (I - \gamma \partial f)$, where $\mathcal{J}_{\gamma \partial g} = (I + \gamma \partial g)^{-1}$, I is an identity operator, γ is a step size. When g is a closed, convex, and proper function, $\mathcal{J}_{\gamma \partial g}$ is equivalent to the well-known proximal operator

$$\text{prox}_{\gamma g}(x) = \arg \min_{y \in \mathbb{X}} (g(y) + \frac{1}{2\gamma} \|y - x\|_2^2),$$

with $\|\cdot\|_2$ denoting the L_2 norm on \mathbb{X} , and, the corresponding algorithm is known as proximal point approach (PPA). More examples are deferred to Table 1.

S-KM Performance with Non-i.i.d. Data

We next study the properties of S-KM iteration when using non-i.i.d. training samples. We show that, under some mild conditions, S-KM iterates concentrate around the true value. These general results are fundamental and cover many splitting algorithms as special cases. Because splitting algorithms are driven by fixed-point operators, it becomes natural to perform the analysis in terms of Fixed Point Residual (FPR) (Davis and Yin 2016), defined as

$$e_k^2 = \|Tx^k - x^k\|^2,$$

which are related to differences between successive KM iterates through $x^{k+1} - x^k = \lambda_k (T(x^k) - x^k)$. In first-order algorithms, with the assumption that, $\epsilon_k = 0, \forall k$, FPR typically relates to the gradient of the objective. For example, in the unite-step gradient descent algorithms $x^k = x^{k-1} - \nabla f(x^{k-1})$, and so the FPR is given by $\|\nabla f(x^{k-1})\|^2$. Thus, FPR convergence naturally implies the convergence of $\|x^{k+1} - x^k\|^2$.

We proceed by establishing the properties of the ergodic FPR. We first, through Theorem 3, provide the following boundedness on the expectation of the norm of the approximation error due to randomness of sampling from P rather than \mathbb{P} .

Assumption 2 \mathbb{X} is compact and has finite radius r : specifically, for any $x, x^* \in \mathbb{X}$, $\|x - x^*\| \leq r < \infty$.

Assumption 2 is the same as the one for online algorithms with correlated data (Agarwal and Duchi 2012; Sun, Sun, and Yin 2018) and common in the online learning, optimization literature.

Theorem 3 (Boundedness on approximation error) Under Assumption 2, the norm of the difference between the true function $T(x^k; \xi) - x^k$ with ξ drawing from \mathbb{P} and its approximation $T(x^k; \xi_{k+1}) - x^k$ with ξ_{k+1} drawing from $\widehat{\mathbb{P}}_K$ is uniformly bounded in expectation. Specifically, $\mathbb{E}\|\epsilon_k\| \leq \Delta$, where

$$\Delta = \left(\frac{8r^2 C(\sum_{k=1}^{\infty} \phi(k))}{K^2} \right)^{\frac{1}{2}} \Gamma\left(\frac{1}{2}\right)$$

and $\Gamma(z) = \int_0^{\infty} x^{z-1} \exp(-x) dx$ is the gamma function.

Theorem 3 suggests that the approximation becomes close to the true one when K increases. However, the noisy can be large when K is small. This is because we are considering drifting distributions and, in the worse case, P^k can be quit far away from \mathbb{P} . Therefore, both Theorem 3 and MR approach indicate that underestimating the mixing time can potentially backfires.

Theorem 4 (Bound of fixed point residual) Let $\bar{e}_K = \Lambda_K^{-1} \sum_{k=1}^K \lambda_k e_k$, with $e_k = Tx^k - x^k$, $\Lambda_K = \sum_{k=1}^K \lambda_k$, and $\lambda_k \in (0, 1)$. Under Assumption 2,

$$\mathbb{E}\|\bar{e}_K\| \leq \frac{2r(1 + \phi(1)) + 2 \sum_{k=1}^K \mathbb{E}\|\lambda_k \epsilon_k\|}{\Lambda_K}.$$

Table 1: Overview of several first-order algorithms

Algorithm	Operator identity	Subgradient identity
SGD ($g = 0$)	$I - \gamma \nabla f$	$x^{k+1} = x^k - \gamma_k \nabla f(x^k)$
PPA ($g = 0$)	$(I + \gamma \partial f)^{-1}$	$x^{k+1} = \text{prox}_{\gamma_k f}(x^k)$
PGD	$(I + \gamma \partial g)^{-1}(I - \gamma \nabla f)$	$x^{k+1} = \text{prox}_{\gamma_k g}(I - \gamma_k \nabla f(x^k))$
DRS	$(I + \gamma \partial f)^{-1} [(I + \gamma \partial g)^{-1} (I - \gamma \partial f) + \gamma \partial f]$	$x^{k+1} = \frac{1}{2}x^k + \frac{1}{2} \text{refl}_{\gamma_k \partial f} \circ \text{refl}_{\gamma_k \partial g}(x^k)$
Relaxed PRS	$(I + \gamma \partial f)^{-1} (I - \lambda \partial g) (I + \lambda \partial g)^{-1} (I - \lambda \partial f)$	$x^{k+1} = (1 - \lambda_k)x^k + \lambda_k \text{refl}_{\gamma_k \partial f} \circ \text{refl}_{\gamma_k \partial g}(x^k)$

When the data is i.i.d. following the distribution \mathbb{P} , then $\phi(1) = 0$. The result in Theorem 4 consequently reduces to that for S-KM iterations with i.i.d. samples. To establish an upper bound for S-KM iterates around the true value, we introduce the following two assumptions.

Assumption 3 *There is a non-increasing sequence $\kappa(k)$ such that, if x^k and x^{k+1} are successive S-KM iterates, then $\mathbb{E}\{\|x^{k+1} - x^k\| \mid \mathcal{F}_t\} \leq \kappa(k)$.*

Assumption 4 *For a sequence of samples ξ_1, \dots, ξ_K , the S-KM iteration produces a sequence of iterates x^1, \dots, x^{K-1} such that $\sum_{k=0}^{K-1} \|T_{\lambda_k}(x^k; \xi_{k+1}) - T_{\lambda_k}(x^*; \xi_{k+1})\| \leq R_{K-1}$.*

Assumption 3 ensures that S-KM iterates are approximately stable. A similar condition for the non-i.i.d. setting is also given in Agarwal and Duchi (2012). Assumption 4 allows us to quantify the impact of our assumptions on the performance of some specific instances of T .

Theorem 5 (*Deviation bound for iterates*) *Under Assumptions 2, 3, and 4, for any $\tau > 0$,*

$$\begin{aligned} \mathbb{E}\left[\left\|\sum_{k=1}^K (x^k - x^*)\right\|\right] &\leq (1 + \phi(1))\mathbb{E}[R_{K-1}] \\ &+ 2(K - \tau)r\sqrt{\phi(\tau + 1)} + \tau\left(\sum_{k=1}^{K-\tau} \mathbb{E}[\kappa(k - 1)] + r\right). \end{aligned}$$

In the case where $\tau = 0$, $\phi(1) = 0$ gives a bound for the i.i.d. setting.

Application

There are many works that focus on using stochastic operator splitting algorithms to solve structured optimization problems (Xu 2020; Rosasco, Villa, and Vü 2019; Yun, Lozano, and Yang 2020; Ouyang et al. 2013). We next present several examples of nonexpansive operators that cover widely used algorithms based on the computation of proximal and gradient operators. For simplicity, we assume that step size $\gamma = \gamma_k$ for all k .

Stochastic PGD Suppose that the function f in problem (4) is convex and differentiable with a $(1/\beta)$ -Lipschitz continuous gradient for some $\beta > 0$ and that $g : \mathbb{X} \rightarrow \mathbb{R} \cup \{\infty\}$ is a proper, closed, lower semi-continuous convex function. Solving problem (4) is equivalent to finding $x \in \mathbb{X}$ such that $0 \in \partial f(x) + \partial g(x)$. Stochastic PGD (S-PGD), due to its simplicity, efficiency, and empirical performance, is commonly

used to solve this problem. For all $k \geq 0$ and $\gamma \in (0, 2\beta)$, the iteration step at time k can be written as

$$x^{k+1} = \text{prox}_{\gamma \partial g}(x^k - \gamma \nabla f(x^k; \xi_{k+1}) + \epsilon_{f,k}) + \epsilon_{g,k}.$$

We will show that the S-PGD algorithm is a special case of the S-KM iteration. Let $T_1 = \mathcal{J}_{\gamma \partial g}$ and $T_2 = (I - \gamma \nabla f)$. Then $T_{\text{PGD}} = T_1 \circ T_2$. Since T_1 is $(1/2)$ -averaged and T_2 is $(\gamma/(2\beta))$ -averaged, it follows that T_{PGD} is $2\beta/(4\beta - \gamma)$ -averaged (Bauschke and Combettes 2011). Since ∇f is single-valued, we have that, for all $k \in \mathbb{N}$ and $x \in \mathbb{X}$,

$$\begin{aligned} x \in (\nabla f + \partial g)^{-1}(0) &\Leftrightarrow x - \gamma \nabla f(x) \in x + \gamma \partial g(x) \\ &\Leftrightarrow x \in \text{Fix}(T_1 \circ T_2). \end{aligned}$$

The results of Theorem 3-5 then hold. The following corollary further establishes a generalized error bound for regret for S-PGD.

Corollary 1 (*Generalized error bound for regret: S-PGD*) *Under Assumption 2 and let $\bar{x} = K^{-1} \sum_{k=1}^K x^k$ and $\underline{\tau} = \inf_k \tau_k$, with $\tau_k = \lambda_k(1 - \lambda_k)$, we have*

$$\begin{aligned} &\mathbb{E}\{f(\bar{x}) + g(\bar{x}) - [f(x^*) + g(x^*)]\} \\ &\leq \frac{r^2}{2K\gamma} + \left(\frac{1}{\beta} - \frac{1}{\gamma}\right) \left(4r^2 + \frac{8r^2\pi C(\sum_{k=1}^{\infty} \phi(k))}{K\underline{\tau}}\right). \end{aligned}$$

Stochastic generalized DRS A line search can be used to guarantee the convergence of the S-PGD algorithm if the Lipschitz constant of ∇f is not known. Finding an appropriate step size, however, presents another expensive practical challenge. We introduce the Stochastic generalized Douglas–Rachford Splitting (S-gDRS) algorithm to avoid choosing the step size altogether. The results given in Theorem 3–5 hold by specifying T in the S-KM iteration as

$$T_{\text{DRS}} = \frac{1}{2}(\text{refl}_{\gamma \partial f} \circ \text{refl}_{\gamma \partial g} + I),$$

where $\text{refl}_{\gamma \partial f}$ and $\text{refl}_{\gamma \partial g}$ are reflection operators.

Definition 3 *Given any operator $T : \mathbb{X} \rightarrow \mathbb{X}$, let $\mathcal{J}_{\gamma T}$ denote the operator $(I + \gamma T)^{-1}$. The operator $\text{refl}_{\gamma T} = 2\mathcal{J}_{\gamma T} - I$ is called the reflection operator of T .*

Because reflection operators are nonexpansive and the composition of nonexpansive operators is nonexpansive (Bauschke and Combettes 2011), we have that $\text{refl}_{\gamma \partial f} \circ \text{refl}_{\gamma \partial g}$ is a nonexpansive operator. Therefore, T_{DRS} is $(1/2)$ -averaged, indicating that the S-gDRS algorithm is a special case of the S-KM algorithm. In addition, the definition of the reflection operator indicates that the step size $\gamma > 0$ will not affect the convergence of T_{DRS} . This is an advantage over T_{PGD} , which instead requires that $\gamma \in (0, 2\beta)$.

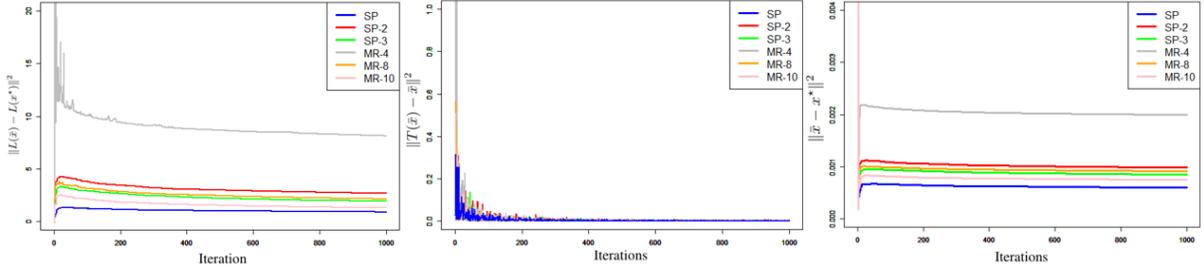


Figure 1: Performance of Algorithm 1 and the other two methods: SP- m and MR- s .

The S-gDRS algorithm (with $\lambda_k = 1, \forall k$) is a special case of the relaxed Peaceman–Rachford Splitting (PRS) algorithm with the iteration,

$$x^{k+1} = (1 - \lambda_k) x^k + \lambda_k \cdot \text{refl}_{\gamma \partial f} \circ \text{refl}_{\gamma \partial g} (x^k)$$

taking $\lambda_k = 1/2, \forall k$. A similar result holds for stochastic relaxed PRS (S-rPRS). We also give the error bound for regret for S-rPRS.

Corollary 2 (*Generalized error bound for regret: S-rPRS*) Under Assumption 2, Let $\bar{x}^f = K^{-1} \sum_{k=1}^K x_k^f$ and $\bar{x}^g = K^{-1} \sum_{k=1}^K x_k^g$ with the auxiliary points $x_k^g = \text{prox}_{\gamma g}(x^k) + \epsilon_g$ and $x_k^f = \text{prox}_{\gamma f}(\text{refl}_{\gamma \partial g}(x^k)) + \epsilon_f$. Denote $\underline{\tau} = \inf_k \tau_k$, with $\tau_k = \lambda_k(1 - \lambda_k)$. We have that,

$$\begin{aligned} & \mathbb{E}[f(\bar{x}^f; \xi) + g(\bar{x}^g; \xi) - (f(x^*; \xi) + g(x^*; \xi))] \\ & \leq \frac{r^2}{4\gamma\lambda K} + \frac{2(\lambda - 1)r^2}{\gamma\lambda^2} + \frac{4r^2\pi}{\gamma\lambda\underline{\tau}} \left(1 - \frac{1}{\lambda}\right) \frac{C(\sum_{k=1}^{\infty} \phi(k))}{K}. \end{aligned}$$

Numerical Experiments We present some numerical results in our setting where samples are generated from an autoregressive process. We show that the iteration procedure in Algorithm 1 does fewer samples and yields better performance than does the multiple replication approach. We also demonstrate the advantage of using each sample of one trajectory in each iteration rather than at regular intervals: although the dependency of data is weakened by using samples at regular intervals, the performance of iteration has not been improved. Our data-generating mechanism resembles that in Duchi et al. (2012). Let A be a subdiagonal matrix with entries $A_{i,i-1} \stackrel{\text{i.i.d.}}{\sim} U[0.8, 0.99]$. We uniformly draw a sparse vector $x \in \mathbb{R}^{1000}$, specifically, with the first non-zero 50 elements of x . The data $\{(\xi_k^1, \xi_k^2)\}_{k \in \mathbb{N}}$ is generated according to the autoregressive process

$$\xi_k^1 = A\xi_{k-1}^1 + e_1 W_k, \quad \xi_k^2 = \langle x, \xi_k^1 \rangle + E_k,$$

where e_1 is the first standard basis vector, the W_k s are i.i.d. $N(0, 1)$ random variables, and the E_k s are i.i.d. biexponential random variables with variances of one. We aim to solve the lasso-type problem

$$\hat{x}_K^* = \underset{x \in \mathbb{R}^d}{\text{argmin}} \left\{ \frac{1}{K} \sum_{k=1}^K \|\langle x, \xi_k^1 \rangle - \xi_k^2\|^2 + \lambda \|x\|_1 \right\},$$

where λ is a pre-set tuning parameter, using PGD. We generate samples from the above autoregressive model in three ways: (SP) samples are the elements of a single trajectory and are used immediately; (SP- m) samples are generated at every m -th element of the same trajectory up to get K samples (e.g., with $K = 4$ and $m = 3$, we keep samples at $k = 1, 4, 7, 10$); and, (MR- s) samples are generated via multiple replication method as the s -th elements of independent trajectories starting from the same state. And, we need simulate K trajectories in total. By not using every element, SP- m weakens dependencies between generated samples. SP and MR are closely resemble the sampling technique in Duchi et al. (2012) and Sun, Sun, and Yin (2018). Given sample size, $K = 1000$, we consider $m = 2, 3$ for SP- m and $s = 4, 6, 8, 10$ for MR- s .

Figure 1 illustrates our numerical results and the convergence behavior of the three methods that is evaluated by three criteria: regret function $L(\bar{x}) - L(x^*)$, FPR and the difference between iterate and true value. As expected, the multiple replication approach shows poor performance for small s as the true mixing time is underestimated: MR-4 has the worst performance under the criteria. Moreover, it becomes clear that using each sample sequentially (SP) rather than attempting to draw weak dependent samples at each iteration from the same trajectory (SP- m) is a more computationally efficient approach.

Conclusion

In this paper, We show that SAA retains its asymptotic consistency and out-of-sample performance when data is not independent and can be solved efficiently in practice, we also evaluate the performance of a class of first-order algorithms and give several examples illustrating the usefulness of our analyses. We provide generalized error bounds for iterates around the true value that show the impact of dependence of the training sample on the convergence result. It may be possible to sharpen these results using monotone operator properties, such as the contraction property of firmly-nonexpansive operators. We leave these investigations to future work.

Acknowledgements

We would like to thank the anonymous reviewers for great feedback on the paper. Dr. Jiang and Dr. Kong were supported by the Natural Sciences and Engineering Re-

search Council of Canada (NSERC). Dr. Kong was also supported by the University of Alberta/Huawei Joint Innovation Collaboration, Huawei Technologies Canada Co., Ltd., and Canada Research Chair in Statistical Learning.

References

- Agarwal, A.; and Duchi, J. C. 2012. The generalization ability of online algorithms for dependent data. *IEEE Transactions on Information Theory*, 59(1): 573–587.
- Bauschke, H. H.; and Combettes, P. L. 2011. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer.
- Bertsimas, D.; Gupta, V.; and Kallus, N. 2018. Robust sample average approximation. *Mathematical Programming*, 171(1): 217–282.
- Boyd, S.; and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.
- Davis, D.; and Drusvyatskiy, D. 2019. Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimization*, 29(1): 207–239.
- Davis, D.; and Yin, W. 2016. Convergence rate analysis of several splitting schemes. In *Splitting methods in communication, imaging, science, and engineering*, 115–163. Springer.
- Dedecker, J.; and Merlevede, F. 2007. The empirical distribution function for dependent variables: asymptotic and nonasymptotic results in L^p . *ESAIM: Probability and Statistics*, 11: 102–114.
- Derman, E.; and Mannor, S. 2020. Distributional robustness and regularization in reinforcement learning. *arXiv preprint arXiv:2003.02894*.
- Duchi, J. C.; Agarwal, A.; Johansson, M.; and Jordan, M. I. 2012. Ergodic mirror descent. *SIAM Journal on Optimization*, 22(4): 1549–1578.
- Dyer, M.; Frieze, A.; Kannan, R.; Kapoor, A.; Perkovic, L.; and Vazirani, U. 1993. A mildly exponential time algorithm for approximating the number of solutions to a multidimensional knapsack problem. *Combinatorics, Probability and Computing*, 2(3): 271–284.
- Emelogu, A.; Chowdhury, S.; Marufuzzaman, M.; Bian, L.; and Eksioğlu, B. 2016. An enhanced sample average approximation method for stochastic optimization. *International Journal of Production Economics*, 182: 230–252.
- Esfahani, P. M.; and Kuhn, D. 2018. Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1): 115–166.
- Fouskakis, D.; and Draper, D. 2002. Stochastic optimization: a review. *International Statistical Review*, 70(3): 315–349.
- Franklin, J. 2005. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2): 83–85.
- Gelman, A.; and Rubin, D. B. 1992. Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4): 457–472.
- Heyman, D. P.; and Sobel, M. J. 2004. *Stochastic models in operations research: stochastic optimization*, volume 2. Courier Corporation.
- Jerrum, M.; and Sinclair, A. 1996. The Markov chain Monte Carlo method: an approach to approximate counting and integration. *Approximation Algorithms for NP-hard problems*, PWS Publishing.
- Johansson, B.; Rabi, M.; and Johansson, M. 2007. A simple peer-to-peer algorithm for distributed optimization in sensor networks. In *2007 46th IEEE Conference on Decision and Control*, 4705–4710. IEEE.
- Johansson, B.; Rabi, M.; and Johansson, M. 2010. A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM Journal on Optimization*, 20(3): 1157–1170.
- Kim, S.; Pasupathy, R.; and Henderson, S. G. 2015. A guide to sample average approximation. *Handbook of Simulation Optimization*, 207–243.
- Kleywegt, A. J.; Shapiro, A.; and Homem-de Mello, T. 2002. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2): 479–502.
- Kushner, H.; and Yin, G. G. 2003. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media.
- Modha, D. S.; and Masry, E. 1996. Minimum complexity regression estimation with weakly dependent observations. *IEEE Transactions on Information Theory*, 42(6): 2133–2145.
- Ouyang, H.; He, N.; Tran, L.; and Gray, A. 2013. Stochastic alternating direction method of multipliers. In *International Conference on Machine Learning*, 80–88. PMLR.
- Pietrosanu, M.; Gao, J.; Kong, L.; Jiang, B.; and Niu, D. 2020. Advanced algorithms for penalized quantile and composite quantile regression. *Computational Statistics*, 1–14.
- Pietrosanu, M.; Shu, H.; Jiang, B.; Kong, L.; Heo, G.; He, Q.; Gilmore, J.; and Zhu, H. 2021. Estimation for the bivariate quantile varying coefficient model with application to diffusion tensor imaging data analysis. *Biostatistics*.
- Rosasco, L.; Villa, S.; and Vũ, B. C. 2019. Convergence of stochastic proximal gradient algorithm. *Applied Mathematics & Optimization*, 1–27.
- Sun, T.; Sun, Y.; and Yin, W. 2018. On markov chain gradient descent. *arXiv preprint arXiv:1809.04216*.
- Teo, C. H.; Vishwanathan, S.; Smola, A.; and Le, Q. V. 2010. Bundle Methods for Regularized Risk Minimization. *Journal of Machine Learning Research*, 11(1).
- Wang, Y.; Kong, L.; Jiang, B.; Zhou, X.; Yu, S.; Zhang, L.; and Heo, G. 2019. Wavelet-based LASSO in functional linear quantile regression. *Journal of Statistical Computation and Simulation*, 89(6): 1111–1130.
- Xu, Y. 2020. Primal-dual stochastic gradient method for convex programs with many functional constraints. *SIAM Journal on Optimization*, 30(2): 1664–1692.

Yu, D.; Zhang, L.; Mizera, I.; Jiang, B.; and Kong, L. 2019. Sparse wavelet estimation in quantile regression with multiple functional predictors. *Computational Statistics & Data Analysis*, 136: 12–29.

Yun, J.; Lozano, A. C.; and Yang, E. 2020. A general family of stochastic proximal gradient methods for deep learning. *arXiv preprint arXiv:2007.07484*.

Zhang, T. 2004. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the Twenty-first International Conference on Machine Learning*, 116.

Zhang, Z.; Wang, X.; Kong, L.; and Zhu, H. 2021. High-Dimensional Spatial Quantile Function-on-Scalar Regression. *Journal of the American Statistical Association*, 1–16.