Doubly Random Parallel Stochastic Algorithms for Large Scale Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider learning problems over training sets in which both, the number of training examples and the dimension of the feature vectors, are large. To solve these problems we propose the random parallel stochastic algorithm (RAPSA). We call the algorithm random parallel because it utilizes multiple processors to operate in a randomly chosen subset of blocks of the feature vector. We call the algorithm parallel stochastic because processors choose elements of the training set randomly and independently. Algorithms that are parallel in either of these dimensions exist, but RAPSA is the first attempt at a methodology that is parallel in both, the selection of blocks and the selection of elements of the training set. In RAPSA, processors utilize the randomly chosen functions to compute the stochastic gradient component associated with a randomly chosen block. The technical contribution of this paper is to show that this minimally coordinated algorithm converges to the optimal classifier when the training objective is convex. In particular, we show that: (i) When using decreasing stepsizes, RAPSA converges almost surely over the random choice of blocks and functions. (ii) When using constant stepsizes, convergence is to a neighborhood of optimality with a rate that is linear in expectation. Acelerated (A)RAPSA is further proposed by leveraging ideas of stochastic quasi-Newton optimization. Both, RAPSA and ARAPSA, are numerically evaluated on the MNIST digit recognition problem.

1 Introduction

Learning is often formulated as an optimization problem that finds a classifier $\mathbf{x}^* \in \mathbb{R}^p$ that minimizes the average of a loss function across the elements of a training set. For a precise definition consider a training set with N elements and let $f_n : \mathbb{R}^p \to \mathbb{R}$ be a convex loss function associated with the nth element of the training set. The optimal classifier $\mathbf{x}^* \in \mathbb{R}^p$ is defined as the minimizer of the average cost $F(\mathbf{x}) := (1/N) \sum_{n=1}^{N} f_n(\mathbf{x})$,

$$\mathbf{x}^* := \operatorname*{argmin}_{\mathbf{x}} F(\mathbf{x}) := \operatorname*{argmin}_{\mathbf{x}} \frac{1}{N} \sum_{n=1}^{N} f_n(\mathbf{x}). \tag{1}$$

Problems such as support vector machines, logistic regression, and matrix completion can be put in the form of problem (1). In this paper we are interested in large scale problems where both, the number of features p and the number of elements N in the training set are very large – which arise, e.g., in text [1], image [2], and genomic [3] processing.

When N and p are large, the parallel processing architecture in Figure 1 becomes of interest. In this architecture, features are divided in B blocks each of which contains $p_b \ll p$ features and a set of $I \ll B$ processors work in paralell on randomly chosen feature blocks while using a stocahstic

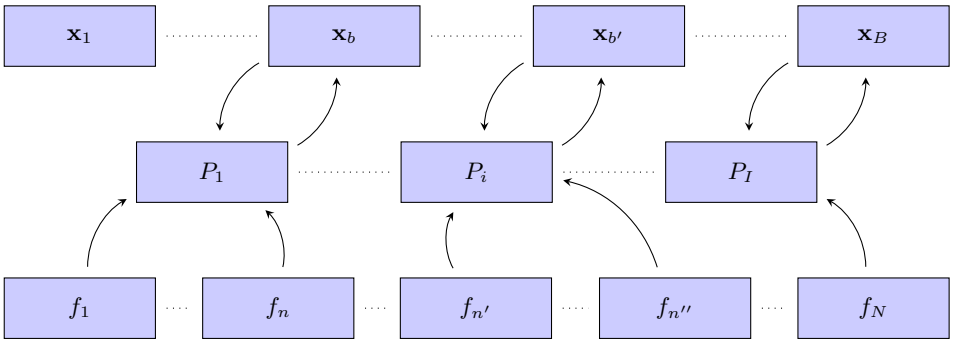


Figure 1: Random parallel stochastic algorithm (RAPSA). At each iteration, processor P_i picks a random block from the set $\{\mathbf{x}_1, \ldots, \mathbf{x}_B\}$ and a random set of functions from the training set $\{f_1, \ldots, f_N\}$. The functions drawn are used to evaluate a stochastic gradient component associated with the chosen block. RAPSA is shown here to converge to the optimal argument $\mathbf{x}^*$ of (1).

subset of elements of the training set. In the schematic shown, Processor 1 fetches functions f_1 and f_n to operate on block $\mathbf{x}_b$ and Processor i fetches functions $f_{n'}$ and $f_{n''}$ to operate on block $\mathbf{x}_{b'}$. Other processors select other elements of the training set and other blocks with the majority of blocks remaining unchanged and the majority of functions remaining unused. The blocks chosen for update and the functions fetched for determination of block updates are selected independently at random in subsequent slots.

Problems that operate on blocks of the feature vectors *or* subsets of the training set, but not on both, blocks *and* subsets, exist. Block coordinate descent (BCD) is the generic name for methods in which the variable space is divided in blocks that are processed separately. Early versions operate by cyclically updating all coordinates at each step [4, 5], while more recent parallelized versions of coordinate descent have been developed to accelerate convergence of BCD [6–8]. Closer to the architecture in Figure 1, methods in which subsets of blocks are selected at random have also been proposed [9]. BCD, serial, parallel, or random, can handle cases where the parameter dimension p is large but requires access to all training samples at each iteration.

Methods that utilize a subset of functions are known by the generic name of stochastic approximation and rely on the use of stochastic gradients. In plain stochastic gradient descent (SGD), the gradient of the aggregate function is estimated by the gradient of a randomly chosen function f_n [10]. Since convergence of SGD is slow more often that not, various recent developments have been aimed at accelerating convergence. These attempts include methodologies to reduce the variance of stochastic gradients [?, SAG, SAGA, SVRG] and the use of ideas from quasi-Newton optimization to handle difficult curvature profiles [11, 12]. More pertinent to the work considered here are the use of cyclic block SGD updates [13] and the exploitation of sparsity properties of feature vectors to allow for parallel updates [14]. These methods are suitable when the number of elements in the training set N is large but don't allow for parallel feature processing unless parallelism is inherent to the problem's structure.

The random parallel stochastic algorithm (RAPSA) proposed in this paper represents the first effort at implementing the architecture in Figure 1 that randomizes over both, features and sample functions. In RAPSA, the functions fetched by a processor are used to compute the stochastic gradient component associated with a randomly chosen block (Section 2). The processors do not coordinate in either choice except to avoid selection of the same block. Our main technical contribution is to show that RAPSA iterates converge to the optimal classifier $\mathbf{x}^*$ when using a sequence of decreasing stepsizes and to a neighborhood of the optimal classifier when using constant stepsizes (Section 2.1). In the latter case, we further show that the rate of convergence to this optimality neighborhood is linear in expectation. These results are interesting because only a subset of features are updated per iteration and the functions used to update different blocks are, in general, different. We further propose an acceleration of RAPSA in which processors also update and exploit a curvature estimation matrix associated with each block (Section 3). This accelerated (A)RAPSA algorithm leverages ideas of stochastic quasi-Newton methods [11, 12] and results in faster convergence. Both, RAPSA and ARAPSA, are numerically evaluated on the MNIST digit recognition problem (Section 4).

2 Random Parallel Stochastic Algorithm (RAPSA)

We consider a more general formulation of (1) in which the number N of functions f_n is not necessarily finite. Introduce then a random variable $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^q$ that determine the choice of the random smooth convex function $f(\cdot, \boldsymbol{\theta}) : \mathbb{R}^p \to \mathbb{R}$. We consider the problem of minimizing the expectation of the random functions $F(\mathbf{x}) := \mathbb{E}_{\boldsymbol{\theta}}[f(\mathbf{x}, \boldsymbol{\theta})]$,

$$\mathbf{x}^* := \operatorname*{argmin}_{\mathbf{x}} F(\mathbf{x}) := \operatorname*{argmin}_{\mathbf{x}} \mathbb{E}_{\boldsymbol{\theta}} \left[f(\mathbf{x}, \boldsymbol{\theta}) \right]. \tag{2}$$

Problem (1) is a particular case of (2) in which each of the functions f_n is drawn with probability $1/N$. We refer to $f(\cdot, \boldsymbol{\theta})$ as instantaneous functions and to $F(\mathbf{x})$ as the average function.

RAPSA utilizes I processors to update a random subset of blocks of the variable $\mathbf{x}$, with each of the blocks relying on a subset of randomly and independently chosen elements of the training set; see Figure 1. Formally, decompose the variable $\mathbf{x}$ into B blocks to write $\mathbf{x} = [\mathbf{x}_1; \ldots; \mathbf{x}_B]$, where block b has length p_b so that we have $\mathbf{x}_b \in \mathbb{R}^{p_b}$. At iteration t, processor i selects a random index b_i^t for updating and a random subset $\boldsymbol{\Theta}_i^t$ of L instantaneous functions. It then uses these instantaneous functions to determine stochastic gradient components for the subset of variables $\mathbf{x}_b = \mathbf{x}_{b_i^t}$ as an average of the components of the gradients of the functions $f(\mathbf{x}^t, \boldsymbol{\theta})$ for $\boldsymbol{\theta} \in \boldsymbol{\Theta}_i^t$,

$$\nabla_{\mathbf{x}_b} f(\mathbf{x}^t, \boldsymbol{\Theta}_i^t) = \frac{1}{L} \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}_i^t} \nabla_{\mathbf{x}_b} f(\mathbf{x}^t, \boldsymbol{\theta}), \qquad b = b_i^t. \tag{3}$$

The stochastic gradient block in (3) is then modulated by a possibly time varying stepsize γ^t and used by processor i to update the block $\mathbf{x}_b = \mathbf{x}_{b_i^t}$

$$\mathbf{x}_b^{t+1} = \mathbf{x}_b^t - \gamma^t \nabla_{\mathbf{x}_b} f(\mathbf{x}^t, \boldsymbol{\Theta}_i^t) \qquad\qquad b = b_i^t. \tag{4}$$

RAPSA is defined by the joint implementation of (3) and (4) across all I processors. The selection of blocks is coordinated so that no processors operate in the same block. The selection of elements of the training set is uncoordinated across processors. The fact that at any point in time a random subset of blocks is being updated utilizing a random subset of elements of the training set means that RAPSA requires almost no coordination between processors. The contribution of this paper is to show that this very lean algorithm converges to the optimal argument $\mathbf{x}^*$ as we show in the following section.

2.1 Convergence Analysis

We show in this section that the sequence of objective function values $F(\mathbf{x}^t)$ generated by RAPSA approaches the optimal objective function value $F(\mathbf{x}^*)$. In establishing this result we define the set $\mathcal{S}^t$ containing the blocks that are updated at step t with associated indices $\mathcal{I}^t \subset \{1, \ldots, B\}$. Note that components of the set $\mathcal{S}^t$ are chosen uniformly at random from the set of blocks $\{\mathbf{x}_1, \ldots, \mathbf{x}_B\}$. The definition of $\mathcal{S}^t$ is such that the time evolution of RAPSA iterates can be written as, [cf. (4)],

$$\mathbf{x}_i^{t+1} \;=\; \mathbf{x}_i^t - \gamma^t \, \nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \boldsymbol{\Theta}_i^t) \qquad \text{for all } \mathbf{x}_i \in \mathcal{S}^t, \tag{5}$$

while the rest of the blocks remain unchanged, i.e., $\mathbf{x}_i^{t+1} = \mathbf{x}_i^t$ for $\mathbf{x}_i \notin \mathcal{S}^t$. Since the number of updated blocks is equal to the number of processors, the ratio of updated blocks is $r := |\mathcal{I}^t|/B = I/B$.

To prove convergence of RAPSA, we require the following assumptions

Assumption 1. *The instantaneous objective functions $f_i(\mathbf{x})$ are differentiable and the average function $F(\mathbf{x})$ is strongly convex with parameter $m > 0$.*

Assumption 2. *The average objective function gradients $\nabla F(\mathbf{x})$ are Lipschitz continuous with respect to the Euclidian norm with parameter M. I.e., for all $\mathbf{x}, \hat{\mathbf{x}} \in \mathbb{R}^p$, it holds*

$$\|\nabla F(\mathbf{x}) - \nabla F(\hat{\mathbf{x}})\| \;\leq\; M \, \|\mathbf{x} - \hat{\mathbf{x}}\|. \tag{6}$$

Assumption 3. The second moment of the norm of the stochastic gradient is bounded for all $\mathbf{x}$, i.e., there exists a constant K such that for all variables $\mathbf{x}$, it holds

$$\mathbb{E}_{\boldsymbol{\theta}} \left[\|\nabla f(\mathbf{x}^t, \boldsymbol{\theta}^t)\|^2 \,\middle|\, \mathbf{x}^t \right] \leq K. \tag{7}$$

Notice that Assumption 1 only enforces strong convexity of the average function F, while the instantaneous functions f_i may not be even convex. Further, notice that since the instantaneous functions f_i are differentiable the average function F is also differentiable. The Lipschitz continuity of the average function gradients ∇F is customary in proving objective function convergence for descent algorithms. The restriction imposed by Assumption 3 is a standard condition in stochastic approximation literature [10], its intent being to limit the variance of the stochastic gradients [15].

Our first result comes in the form of a expected descent lemma that relates the expected difference of subsequent iterates to the gradient of the average function.

Lemma 1. *Consider the random parallel stochastic algorithm defined in* (3)-(5)*. Recall the definitions of the set of updated blocks $\mathcal{I}^t$ which are randomly chosen from the total B blocks. Define $\mathcal{F}^t$ as a sigma algebra that measures the history of the system up until time t. Then, the expected value of the difference $\mathbf{x}^{t+1} - \mathbf{x}^t$ with respect to the random set $\mathcal{I}^t$ given $\mathcal{F}^t$ is*

$$\mathbb{E}_{\mathcal{I}^t}\left[\mathbf{x}^{t+1} - \mathbf{x}^t \mid \mathcal{F}^t\right] = -r\gamma^t \, \nabla f(\mathbf{x}^t, \mathbf{\Theta}^t). \tag{8}$$

Moreover, the expected value of the squared norm $\|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2$ with respect to the random set $\mathcal{S}^t$ given $\mathcal{F}^t$ can be simplified as

$$\mathbb{E}_{\mathcal{I}^t}\left[\|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 \mid \mathcal{F}^t\right] = r(\gamma^t)^2 \, \left\|\nabla f(\mathbf{x}^t, \mathbf{\Theta}^t)\right\|^2. \tag{9}$$

Notice that in the regular stochastic gradient descent method the difference of two consecutive iterates $\mathbf{x}^{t+1} - \mathbf{x}^t$ is equal to the stochastic gradient $\nabla f(\mathbf{x}^t, \mathbf{\Theta}^t)$ times the stepsize γ^t. According to the first result in Lemma 1, the expected value of stochastic gradients with respect to the random set of blocks $\mathcal{I}^t$ is the same as the one for SGD except that it is multiplied by the fraction of updated blocks r. Expression in (9) shows the same relation for the expected value of the squared difference $\|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2$. These relationships confirm that in expectation RAPSA behaves as SGD which allows us to establish the global convergence of RAPSA.

Proposition 1. *Consider the random parallel stochastic algorithm defined in* (3)- (5)*. If Assumptions 1-3 hold true, then the objective function error sequence $F(\mathbf{x}^t) - F(\mathbf{x}^*)$ satisfies*

$$\mathbb{E}\left[F(\mathbf{x}^{t+1}) - F(\mathbf{x}^*) \mid \mathcal{F}^t\right] \leq \left(1 - 2mr\gamma^t\right)\left(F(\mathbf{x}^t) - F(\mathbf{x}^*)\right) + \frac{rMK(\gamma^t)^2}{2}. \tag{10}$$

Proposition 1 leads to a supermartingale relationship for the sequence of objective function errors $F(\mathbf{x}^t) - F(\mathbf{x}^*)$. In the following theorem we show that if the sequence of stepsize satisfies standard stochastic approximation diminishing step-size rules (non-summable and squared summable), the sequence of objective function errors $F(\mathbf{x}^t) - F(\mathbf{x}^*)$ converges to null almost surely. Considering the strong convexity assumption this result implies almost sure convergence of the sequence $\|\mathbf{x}^t - \mathbf{x}^*\|^2$ to null.

Theorem 1. *Consider the random parallel stochastic algorithm defined in* (3)-(5)*. If Assumptions 1-3 hold true and the sequence of stepsizes are non-summable $\sum_{t=0}^{\infty} \gamma^t = \infty$ and square summable $\sum_{t=0}^{\infty} (\gamma^t)^2 < \infty$, then sequence of the variables $\mathbf{x}^t$ generated by RAPSA converges almost surely to the optimal argument $\mathbf{x}^*$,*

$$\lim_{t\to\infty} \|\mathbf{x}^t - \mathbf{x}^*\|^2 \; = \; 0 \qquad a.s. \tag{11}$$

Moreover, if stepsize is defined as $\gamma^t := \gamma^0 T^0/(t+T^0)$ and the stepsize parameters are chosen such that $2mr\gamma^0 T^0 > 1$, then the expected average function error $\mathbb{E}\left[F(\mathbf{x}^t) - F(\mathbf{x}^*)\right]$ converges to null at least with a sublinear convergence rate of order $O(1/t)$,

$$\mathbb{E}\left[F(\mathbf{x}^t) - F(\mathbf{x}^*)\right] \; \leq \; \frac{C}{t + T^0}, \tag{12}$$

where the constant C is defined as

$$C = \max\left\{\frac{rMK(\gamma^0 T^0)^2}{4mr\gamma^0 T^0 - 2}, \; T^0(F(\mathbf{x}^0) - F(\mathbf{x}^*))\right\}. \tag{13}$$

The result in Theorem 1 shows that when the sequence of stepsize is diminishing as $\gamma^t = \gamma^0 T^0/(t + T^0)$, the average objective function value $F(\mathbf{x}^t)$ sequence converges to the optimal objective value

$F(\mathbf{x}^*)$ with probability 1.[1] Further, the rate of convergence in expectation is at least in the order of $O(1/t)$. Diminishing stepsizes are useful when exact convergence is required, however, for the case that we are interested in a specific accuracy ϵ the more efficient choice is using a constant stepsize. In the following theorem we study the convergence properties of RAPSA for a constant stepsize $\gamma^t = \gamma$.

Theorem 2. *Consider the random parallel stochastic algorithm defined in* (3) *and* (5)*. If Assumptions 1-3 hold true and the stepsize is constant $\gamma^t = \gamma$, then the sequence of the variables $\mathbf{x}^t$ generated by RAPSA converges almost surely to a neighborhood of the optimal argument $\mathbf{x}^*$ as*

$$\liminf_{t\to\infty} \; F(\mathbf{x}^t) - F(\mathbf{x}^*) \; \leq \; \frac{\gamma M K}{4m} \qquad a.s. \tag{14}$$

Moreover, if the constant stepsize γ is chosen such that $2mr\gamma < 1$ then the expected average function value error $\mathbb{E}\left[F(\mathbf{x}^t) - F(\mathbf{x}^*)\right]$ converges linearly to an error bound as

$$\mathbb{E}\left[F(\mathbf{x}^t) - F(\mathbf{x}^*)\right] \leq (1 - 2m\gamma r)^t \left(F(\mathbf{x}^0) - F(\mathbf{x}^*)\right) + \frac{\gamma M K}{4m}. \tag{15}$$

Notice that according to the result in (15) there exits a trade-off between accuracy and speed of convergence. Decreasing the constant stepsize γ leads to a smaller error bound $\gamma MK/4m$ and a more accurate convergence, while the linear convergence constant $(1 - 2m\gamma r)$ increases and the convergence rate becomes slower. Further, note that the error of convergence $\gamma MK/4m$ is independent of the ratio of updated blocks r, while the constant of linear convergence $1 - 2m\gamma r$ depends on r. Therefore, updating a fraction of the blocks at each iteration decreases the speed of convergence for RAPSA relative to SGD that updates all of the blocks, however, both of the algorithms reach the same accuracy.

To achieve accuracy ϵ the sum of two terms in the right hand side of (15) should be smaller than ϵ. Let's consider ϕ as a positive constant that is strictly smaller than 1, i.e., $0 < \phi < 1$. Then, we want to have

$$\frac{\gamma M K}{4m} \leq \phi\epsilon \;, \qquad (1 - 2m\gamma r)^t \left(F(\mathbf{x}^0) - F(\mathbf{x}^*)\right) \leq (1 - \phi)\epsilon. \tag{16}$$

Therefore, to satisfy the first condition in (16) we set the stepsize as $\gamma = 4m\phi\epsilon/MK$. Apply this substitution into the second inequality in (16) and consider the inequality $a + \ln(1 - a) < 0$ for $0 < a < 1$, to obtain that

$$t \geq \frac{MK}{8m^2 r\phi\epsilon} \ln\left(\frac{F(\mathbf{x}^0) - F(\mathbf{x}^*)}{(1 - \phi)\epsilon}\right). \tag{17}$$

The lower bound in (17) shows the minimum number of required iterations for RAPSA to achieve accuracy ϵ.

3 Accelerated Random Parallel Stochastic Algorithm (ARAPSA)

As we mentioned in Section 2, RAPSA operates on first-order information which may lead to slow convergence in ill-conditioned problems. We introduce Accelerated RAPSA (ARAPSA) as a parallel doubly stochastic algorithm that incorporates second-order information of the objective by separately approximating the function curvature for different blocks. We do this by implementing the oLBFGS algorithm for different blocks of the variable $\mathbf{x}$. Define $\hat{\mathbf{B}}_i^t$ as an approximation for the Hessian inverse of the objective function that corresponds to the block $\mathbf{x}_i$. The update of ARAPSA is defined as multiplication of the descent direction of RAPSA by $\hat{\mathbf{B}}_i^t$, i.e.

$$\mathbf{x}_i^{t+1} \; = \; \mathbf{x}_i^t - \gamma^t \, \hat{\mathbf{B}}_i^t \, \nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \mathbf{\Theta}_i^t) \qquad \text{for all } \mathbf{x}_i \in \mathcal{S}^t. \tag{18}$$

We next detail how to properly specify the block approximate Hessian $\hat{\mathbf{B}}_i^t$ so that it behaves in a manner comparable to the true Hessian. To do so, define for each block coordinate $\mathbf{x}_i$ at step t the variable variation $\mathbf{v}_i^t$ and the stochastic gradient variation $\hat{\mathbf{r}}_i^t$ as

$$\mathbf{v}_i^t \; = \; \mathbf{x}_i^{t+1} - \mathbf{x}_i^t, \qquad \hat{\mathbf{r}}_i^t \; = \; \nabla_{\mathbf{x}_i} f(\mathbf{x}^{t+1}, \mathbf{\Theta}_i^t) - \nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \mathbf{\Theta}_i^t). \tag{19}$$

[1]The expectation on the left hand side of (12), and throughout the subsequent convergence rate analysis, is taken with respect to the algorithm history $\mathcal{F}_0$, which contains all randomness in both $\mathbf{\Theta}_t$ and $\mathcal{I}_t$ for all $t \geq 0$.

Observe that the stochastic gradient variation $\hat{\mathbf{r}}_i^t$ is defined as the difference of stochastic gradients at times $t+1$ and t corresponding to the block $\mathbf{x}_i$ for a common set of realizations $\boldsymbol{\Theta}_i^t$. The term $\nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \boldsymbol{\Theta}_i^t)$ is the same as the stochastic gradient used at time t in (18), while $\nabla_{\mathbf{x}_i} f(\mathbf{x}^{t+1}, \boldsymbol{\Theta}_i^t)$ is computed only to determine the stochastic gradient variation $\hat{\mathbf{r}}_i^t$. An alternative and perhaps more natural definition for the stochastic gradient variation is $\nabla_{\mathbf{x}_i} f(\mathbf{x}^{t+1}, \boldsymbol{\Theta}_i^{t+1}) - \nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \boldsymbol{\Theta}_i^t)$. However, as pointed out in [11], this formulation is insufficient for establishing the convergence of stochastic quasi-Newton methods. We proceed to developing a block-coordinate quasi-Newton method by first noting an important property of the true Hessian, and design our approximate scheme to satisfy this property. In particular, observe that the true Hessian inverse $(\mathbf{H}_i^t)^{-1}$ corresponding to block $\mathbf{x}_i$ satisfies the block secant condition, stated as $(\mathbf{H}_i^t)^{-1}\mathbf{v}_i^t = \hat{\mathbf{r}}_i^t$ when the iterates $\mathbf{x}_i^t$ and $\mathbf{x}_i^{t+1}$ are close to each other. The secant condition may be interpreted as stating that the stochastic gradient of a quadratic approximation of the objective function evaluated at the next iteration agrees with the stochastic gradient at the current iteration. We select a Hessian inverse approximation matrix associated with block $\mathbf{x}_i$ such that it satisfies the secant condition $\hat{\mathbf{B}}_i^{t+1}\mathbf{v}_i^t = \hat{\mathbf{r}}_i^t$, and thus behaves in a comparable manner to the true block Hessian.

The oLBFGS Hessian inverse update rule maintains the secant condition at each iteration by using information of the last $\tau \geq 1$ pairs of variable and stochastic gradient variations $\{\mathbf{v}_i^u, \hat{\mathbf{r}}_i^u\}_{u=t-\tau}^{t-1}$. To state the update rule of oLBFGS for revising the Hessian inverse approximation matrices of the blocks, define a matrix as $\hat{\mathbf{B}}_i^{t,0} := \eta_i^t \mathbf{I}$ for each block i and t, where the constant η_i^t for $t > 0$ is given by

$$\eta_i^t := \frac{(\mathbf{v}_i^{t-1})^T \hat{\mathbf{r}}_i^{t-1}}{\|\hat{\mathbf{r}}_i^{t-1}\|^2}, \tag{20}$$

while the initial value is $\eta_i^t = 1$. The matrix $\hat{\mathbf{B}}_i^{t,0}$ is the initial approximate for the Hessian inverse associated with block $\mathbf{x}_i$. The approximate matrix $\hat{\mathbf{B}}_i^t$ is computed by updating the initial matrix $\hat{\mathbf{B}}_i^{t,0}$ using the last τ pairs of curvature information $\{\mathbf{v}_i^u, \hat{\mathbf{r}}_i^u\}_{u=t-\tau}^{t-1}$. We define the approximate Hessian inverse $\hat{\mathbf{B}}_i^t = \hat{\mathbf{B}}_i^{t,\tau}$ corresponding to block $\mathbf{x}_i$ at step t as the outcome of τ recursive applications of the update

$$\hat{\mathbf{B}}_i^{t,u+1} = (\hat{\mathbf{Z}}_i^{t-\tau+u})^T \hat{\mathbf{B}}_i^{t,u} (\hat{\mathbf{Z}}_i^{t-\tau+u}) + \hat{\rho}_i^{t-\tau+u} (\mathbf{v}_i^{t-\tau+u})(\mathbf{v}_i^{t-\tau+u})^T, \tag{21}$$

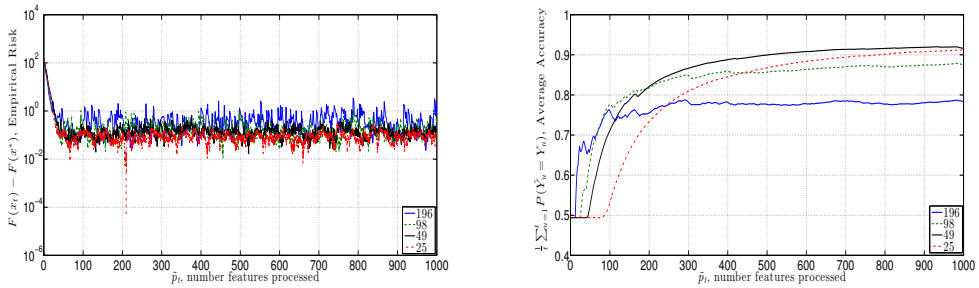
where the matrices $\hat{\mathbf{Z}}_i^{t-\tau+u}$ and the constants $\hat{\rho}_i^{t-\tau+u}$ in (21) for $u = 0, \ldots, \tau-1$ are defined as

$$\hat{\rho}_i^{t-\tau+u} = \frac{1}{(\mathbf{v}_i^{t-\tau+u})^T \hat{\mathbf{r}}_i^{t-\tau+u}} \quad \text{and} \quad \hat{\mathbf{Z}}_i^{t-\tau+u} = \mathbf{I} - \hat{\rho}_i^{t-\tau+u} \hat{\mathbf{r}}_i^{t-\tau+u} (\mathbf{v}_i^{t-\tau+u})^T. \tag{22}$$

The computation cost of $\hat{\mathbf{B}}_i^t$ in (21) is in the order of $O(p_i^2)$, however, for the update in (18) the descent direction $\hat{\mathbf{d}}_i^t := \hat{\mathbf{B}}_i^t \nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \boldsymbol{\Theta}_i^t)$ is required. [16] introduces an efficient implementation of product $\hat{\mathbf{B}}_i^t \nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \boldsymbol{\Theta}_i^t)$ that requires computation complexity of order $O(\tau p_i)$. We use the same idea for computing the descent direction of ARAPSA for each block – more details are provided in supplementary materials. Therefore, the computation complexity of updating each block for ARAPSA is in the order of $O(\tau p_i)$, while RAPSA requires $O(p_i)$ operations. On the other hand, ARAPSA accelerates the convergence of RAPSA by incorporating the second order information of the objective function for the block updates, as may be observed in the numerical analyses provided in Section 4.

4 Numerical analysis

In this section we study the practical performance of the doubly stochastic approximation algorithms developed in Sections 2 and 3 by considering the problem of developing an automated decision system to distinguish between distinct hand-written digits. For that purpose, let $\mathbf{z} \in \mathbb{R}^p$ be a feature vector encoding pixel intensities of an image, and let $y \in \{-1, 1\}$ be an indicator variable of whether the image contains the digit 0 or 8, in which case the binary indicator is respectively $y = -1$ or $y = 1$. We model the task of learning a hand-written digit detector as a logistic regression problem, where one aims to train a classifier $\mathbf{x} \in \mathbb{R}^p$ to determine the relationship between feature vectors $\mathbf{z}_n \in \mathbb{R}^p$ and their associated labels $y_n \in \{-1, 1\}$ for $n = 1, \ldots, N$. The instantaneous functions f_n in (1) may be written as the log-likelihood of a generalized linear model of the odds ratio of



(a) Empirical risk $F(\mathbf{x}^t) - F(\mathbf{x}^*)$ vs. iteration $\tilde{p}_t$ (b) Classification accuracy vs. iteration $\tilde{p}_t$

Figure 2: RAPSA on MNIST data with constant step-size $\gamma = 10^{-1}$, with no mini-batching $L = 1$. Algorithm performance is comparable across different numbers of decision variable coordinates updated per iteration. RAPSA is I times faster than SGD and achieves comparable performance. In some cases, updating fewer variables per iteration improves accuracy.

whether the label is $y_n = 1$ or $y_n = -1$. The empirical risk minimization associated with training set $\mathcal{T} = \{(\mathbf{z}_n, y_n)\}_{n=1}^N$ is to find $\mathbf{x}^*$ as the maximum likelihood estimate,

$$\mathbf{x}^* := \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^p} F(\mathbf{x}) = \frac{\lambda}{2}\|\mathbf{x}\|^2 + \frac{1}{N}\sum_{n=1}^{N} \log(1 + \exp(-y_n \mathbf{x}^T \mathbf{z}_n)) \,. \tag{23}$$

We use the MNIST dataset [17], in which feature vectors $\mathbf{z}_n \in \mathbb{R}^p$ are $p = 28 \times 28$ pixel images whose values are recorded as intensities, or elements of the unit interval $[0, 1]$. Considered here is the subset associated with digits 0 and 8, a training set $\mathcal{T} = \{\mathbf{z}_n, y_n\}_{n=1}^N$ with $N = 1.76 \times 10^4$ sample points. Further, the optimal classifier $\mathbf{x}^*$ in (23) is computed using Liblinear [18].

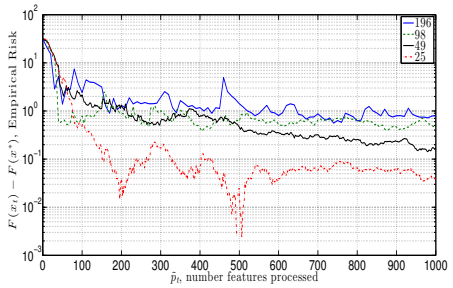
We run RAPSA on this training subset for the case that $B = [p/4] = 196$, where $[\cdot]$ denotes rounding a number to its nearest integer. Moreover, we uniformly allocate feature vectors into blocks of size $p_i = 4$ for $i = 1, \ldots, B$. To determine the advantages of incomplete randomized parallel processing, we vary $|\mathcal{I}^t| = I$, the number of blocks updated at each iteration, set to the number of processors for simplicity, as $I = B$, $I = [B/2] = 98$, $I = [B/4] = 49$, $I = [B/8] = 25$. When $I = B$, RAPSA is parallel stochastic gradient descent. In the subsequent experiment, we set $L = 1$ (no mini-batching).

Comparing algorithm performance over iteration t across varying numbers of blocks updates $|\mathcal{I}^t|$ is unfair. If RAPSA is run on a problem for which $|\mathcal{I}^t| = B/2$, then at iteration t it has only processed *half* the data that parallel SGD has processed by the same iteration. Instead we consider the algorithm performance as compared with the amount of features processed up to the current time $\tilde{p}_t$. Observe that for a feature vector of length p, tp features have been processed by iteration t when all decision variable coordinates are updated at each iteration, as in the case of SGD where $r = 1$. When $r < 1$, $\tilde{p}_t$ must be scaled by the proportion of coordinates updated per iteration, which since blocks are uniformly sized, is prt. Moreover, when the mini-batch size $L > 1$, $\tilde{p}_t = prtL$.

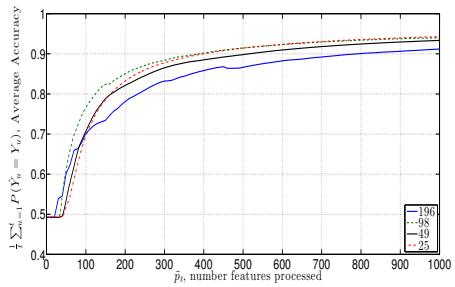
In Figure 2 we show the result of running RAPSA with constant step-size $\gamma^t = 10^{-1}$. In Figure 2(a), we plot $F(\mathbf{x}^t) - F(\mathbf{x}^*)$ versus the number of features processed $\tilde{p}_t$. As in Theorem 2, we observe in Figure 2(a) that the empirical risk $F(\mathbf{x}^t) - F(\mathbf{x}^*)$ converges to a small positive constant of approximately 10^{-1} and the convergence rate difference to a neighborhood of the optimum is negligible. In Figure 2(b) we plot the empirical average classification accuracy on a test set of size $\tilde{N} = 5.88 \times 10^4$. Note that RAPSA with fewer blocks updated (processors) per iteration achieves improved classification accuracy, i.e. $I = B$, $I = [B/2]$, $I = [B/4]$, $I = [B/8]$ respectively achieve accuracies of 78%, 87%, 91%, 92%.

We now run Accelerated RAPSA, or ARAPSA as stated in Section 3 for this problem setting for the entire MNIST binary training subset associated with digits 0 and 8, with mini-batch size $L = 10$ and the level of curvature information set as $\tau = 10$. We select decreasing step-size $\gamma^t = \gamma^0 T^0/(t+T^0)$ with annealing rate $T^0 = [2N/p] = 45$, regularizer $\lambda = 1/\sqrt{N} = 7.5 \times 10^{-3}$, and $\gamma^0 = 2/(\lambda T_0^2)$.

As before, we study the advantages of incomplete randomized parallel processing by varying $I = |\mathcal{I}^t|$, the number of blocks updated at per iteration, as $I \in \{B, [B/2], [B/4], [B/8]\}$. Fig-

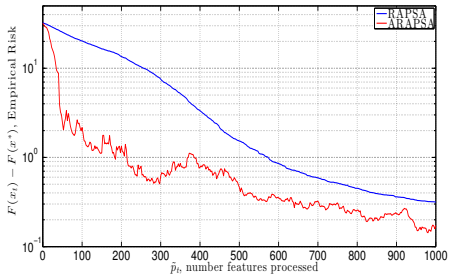


(a) Empirical risk $F(\mathbf{x}^t) - F(\mathbf{x}^*)$ vs. iteration $\tilde{p}_t$

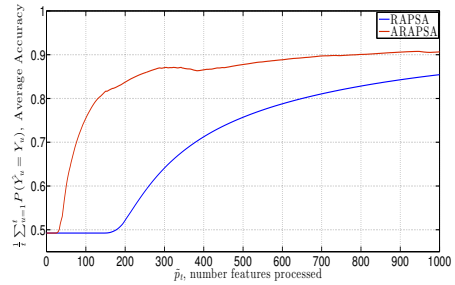


(b) Classification accuracy vs. iteration $\tilde{p}_t$

Figure 3: ARAPSA on MNIST data with regularizer $\lambda = 1/\sqrt{N} = 7.5 \times 10^{-3}$, mini-batch size $L = 10$, curvature information level $\tau = 10$, and diminishing step-size $\gamma^t = \gamma^0 T^0/(t + T^0)$ with annealing rate $T^0 = [2N/p] = 45$. ARAPSA convergence properties hold in practice.



(a) Empirical risk $F(\mathbf{x}^t) - F(\mathbf{x}^*)$ vs. iteration $\tilde{p}_t$



(b) Classification accuracy vs. iteration $\tilde{p}_t$

Figure 4: RAPSA and ARAPSA algorithms on MNIST data with mini-batch size $L = 10$, the level of curvature information set to $\tau = 10$, and constant step-size $\gamma = 10^{-1}$. ARAPSA successfully accelerates the convergence of RAPSA, and achieves higher accuracy per features processed $\tilde{p}_t$.

ure 3(a) shows the error $F(\mathbf{x}^t) - F(\mathbf{x}^*)$ versus the number of features processed $\tilde{p}_t$. Observe that the algorithm achieves convergence across the differing numbers of parallel computing nodes $|\mathcal{I}^t|$, with improved convergence with smaller $|\mathcal{I}^t|$, i.e. for $|\mathcal{I}^t| = B$, $|\mathcal{I}^t| = [B/2]$, $|\mathcal{I}^t| = [B/4]$, $|\mathcal{I}^t| = [B/8]$, the algorithm respectively achieves $F(\mathbf{x}^t) - F(\mathbf{x}^*) \leq 1$ by $\tilde{p}_t = 170$, $\tilde{p}_t = 40$, $\tilde{p}_t = 185$, and $\tilde{p}_t = 77$. Moreover, in Figure 3(b) we observe ARAPSA achieves comparable accuracy across the different levels of block variables updated per iteration, surpassing 90%.

We study the effect of incorporating second-order information into the block-updates. To do so, we fix mini-batch size as $L = 10$ and run ARAPSA and RAPSA for this problem instance with constant step-size $\gamma = 10^{-1}$. Moreover, only a quarter of the blocks per iteration are updated per iteration $|\mathcal{I}^t| = [B/4]$. The results of this experiment are given in Figure 4. In Figure 4(a) we plot $F(\mathbf{x}^t) - F(\mathbf{x}^*)$ versus the number of features processed $\tilde{p}_t$. Observe that ARAPSA converges more quickly and to a point closer to the optimum than RAPSA, i.e. for the benchmark $F(\mathbf{x}^t) - F(\mathbf{x}^*) \leq 1$, ARAPSA requires $\tilde{p}_t = 185$ while RAPSA requires $\tilde{p}_t = 575$ features. This accelerated behavior is also apparent in Figure 4(b) where ARAPSA achieves an accuracy of 80% by $\tilde{p}_t = 130$, whereas RAPSA requires $\tilde{p}_t = 650$ features for the same benchmark.

The comparable performance of RAPSA to SGD in Figure 2(a), and ARAPSA to oLBFGS in Figure 3, demonstrates the practical utility of the proposed method. Because RAPSA may be implemented in parallel, it may achieve the same empirical result as SGD but at a rate of I times faster than SGD. Moreover, in some cases RAPSA achieves superior classification accuracy with fewer blocks updated per iteration. A natural question, given the speedup made possible by parallel computing, is why to select $I < B$. For situations where p is very large, this would require a large number of computing nodes, which may or may not be available.

References

[1] G. Sampson, R. Haigh, and E. Atwell, "Natural language analysis by stochastic optimization: A progress report on project april," *J. Exp. Theor. Artif. Intell.*, vol. 1, no. 4, pp. 271–287, Oct. 1990. [Online]. Available: http://dx.doi.org/10.1080/09528138908953710

[2] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *The Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[3] M. Taşan, G. Musso, T. Hao, M. Vidal, C. A. MacRae, and F. P. Roth, "selecting causal genes from genome-wide association studies via functionally coherent subnetworks," *Nature methods*, 2014.

[4] P. Tseng and C. O. L. Mangasarian, "Convergence of a block coordinate descent method for nondifferentiable minimization," *J. Optim Theory Appl*, pp. 475–494, 2001.

[5] Y. Xu and W. Yin, "A globally convergent algorithm for nonconvex optimization based on block coordinate update," *arXiv preprint arXiv:1410.1386*, 2014.

[6] P. Richtárik and M. Takáč, "Parallel coordinate descent methods for big data optimization," *arXiv preprint arXiv:1212.0873*, 2012.

[7] Z. Lu and L. Xiao, "On the complexity analysis of randomized block-coordinate descent methods," *Mathematical Programming*, pp. 1–28, 2013.

[8] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.

[9] J. Liu, S. J. Wright, C. Ré, V. Bittorf, and S. Sridhar, "An asynchronous parallel stochastic coordinate descent algorithm," *arXiv preprint arXiv:1311.1873*, 2013.

[10] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 09 1951. [Online]. Available: http://dx.doi.org/10.1214/aoms/1177729586

[11] N. Schraudolph, J. Yu, and S. Günter, "A stochastic quasi-newton method for online convex optimization," 2007.

[12] A. Bordes, L. Bottou, and P. Gallinari, "Sgd-qn: Careful quasi-newton stochastic gradient descent," *The Journal of Machine Learning Research*, vol. 10, pp. 1737–1754, 2009.

[13] Y. Xu and W. Yin, "Block stochastic gradient iteration for convex and nonconvex optimization," *ArXiv preprint 1408.2597v2*, Aug. 2014.

[14] F. Niu, B. Recht, C. R, and S. J. Wright, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *In NIPS*, 2011.

[15] A. Nemirovski, A. Juditsky, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on optimization*, vol. 19, no. 4, pp. 1574–1609, 2009.

[16] L. Dong C. and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, no. 45(1-3), pp. 503–528, 1989.

[17] Y. Lecun and C. Cortes, "The MNIST database of handwritten digits." [Online]. Available: http://yann.lecun.com/exdb/mnist/

[18] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

A Proof of Lemma 1

Recall that the components of vector $\mathbf{x}^{t+1}$ are equal to the components of $\mathbf{x}^t$ for the coordinates that are not updated at step t, i.e., $i \notin \mathcal{I}^t$. For the updated coordinates $i \in \mathcal{I}^t$ we know that $\mathbf{x}_i^{t+1} = \mathbf{x}_i^t - \gamma^t \nabla_{\mathbf{x}_i^t} f(\mathbf{x}^t, \boldsymbol{\theta}^t)$. Therefore, $B-I$ blocks of the vector $\mathbf{x}^{t+1} - \mathbf{x}^t$ are 0 and the remaining I randomly chosen blocks are given by $-\gamma^t \nabla_{\mathbf{x}_i^t} f(\mathbf{x}^t, \boldsymbol{\theta}^t)$. Notice that there are $\binom{B}{I}$ different ways for picking I blocks out of the whole B blocks. Therefore, the probability of each combination of blocks is $1/\binom{B}{I}$. Further, each block appears in $\binom{B-1}{I-1}$ of the combinations. Therefore, the expected value can be written as

$$\mathbb{E}_{\mathcal{I}^t}\left[\mathbf{x}^{t+1} - \mathbf{x}^t \mid \mathcal{F}^t\right] = \frac{\binom{B-1}{I-1}}{\binom{m}{I}} \left(-\gamma^t \nabla f(\mathbf{x}^t, \boldsymbol{\Theta}^t)\right) \tag{24}$$

Observe that simplifying the ratio in the right hand sides of (24) leads to

$$\frac{\binom{B-1}{I-1}}{\binom{B}{I}} = \frac{\frac{(B-1)!}{(I-1)! \times (B-I)!}}{\frac{p!}{I! \times (B-I)!}} = \frac{I}{B} = r. \tag{25}$$

Substituting the simplification in (25) into (24) follows the claim in (8). To prove the claim in (9) we can use the same argument that we used in proving (8) to show that

$$\mathbb{E}_{\mathcal{I}^t}\left[\|\mathbf{x}_{t+1} - \mathbf{x}^t\|^2 \mid \mathcal{F}^t\right] = \frac{\binom{B-1}{I-1}}{\binom{B}{I}} (\gamma^t)^2 \left\|\nabla f(\mathbf{x}^t, \boldsymbol{\Theta}^t)\right\|^2. \tag{26}$$

By substituting the simplification in (25) into (26) the claim in (9) follows.

B Proof of Proposition 1

By considering the Taylor's expansion of $F(\mathbf{x}^{t+1})$ near the point $\mathbf{x}^t$ and observing the Lipschitz continuity of gradients ∇F with constant M we obtain that the average objective function $F(\mathbf{x}^{t+1})$ is bounded above by

$$F(\mathbf{x}^{t+1}) \leq F(\mathbf{x}^t) + \nabla F(\mathbf{x}^t)^T (\mathbf{x}^{t+1} - \mathbf{x}^t) + \frac{M}{2} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2. \tag{27}$$

Compute the expectation of the both sides of (27) with respect to the random set $\mathcal{I}^t$ given the observed set of information $\mathcal{F}^t$. Substitute $\mathbb{E}_{\mathcal{I}^t}\left[\mathbf{x}^{t+1} - \mathbf{x}^t \mid \mathcal{F}^t\right]$ and $\mathbb{E}_{\mathcal{I}^t}\left[\|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 \mid \mathcal{F}^t\right]$ with their simplifications in (8) and (9), respectively, to write

$$\mathbb{E}_{\mathcal{I}^t}\left[F(\mathbf{x}^{t+1}) \mid \mathcal{F}^t\right] \leq F(\mathbf{x}^t) - r\gamma^t \, \nabla F(\mathbf{x}^t)^T \nabla f(\mathbf{x}^t, \boldsymbol{\Theta}^t) + \frac{rM(\gamma^t)^2}{2} \left\|\nabla f(\mathbf{x}^t, \boldsymbol{\Theta}^t)\right\|^2. \tag{28}$$

Notice that the stochastic gradient $\nabla f(\mathbf{x}^t, \boldsymbol{\Theta}^t)$ is an unbiased estimate of the average function gradient $\nabla F(\mathbf{x}^t)$. Therefore, we obtain $\mathbb{E}_{\boldsymbol{\Theta}^t}\left[\nabla f(\mathbf{x}^t, \boldsymbol{\Theta}^t) \mid \mathcal{F}^t\right] = \nabla F(\mathbf{x}^t)$. Observing this relation and considering the assumption in (7), the expected value of (28) with respect to the set of realizations $\boldsymbol{\Theta}^t$ can be written as

$$\mathbb{E}_{\mathcal{I}^t, \boldsymbol{\Theta}^t}\left[F(\mathbf{x}^{t+1}) \mid \mathcal{F}^t\right] \leq F(\mathbf{x}^t) - r\gamma^t \, \left\|\nabla F(\mathbf{x}^t)\right\|^2 + \frac{rM(\gamma^t)^2 K}{2}. \tag{29}$$

Subtracting the optimal objective function value $F(\mathbf{x}^*)$ form the both sides of (29) implies that

$$\mathbb{E}_{\mathcal{I}^t, \boldsymbol{\Theta}^t}\left[F(\mathbf{x}^{t+1}) - F(\mathbf{x}^*) \mid \mathcal{F}^t\right] \leq F(\mathbf{x}^t) - F(\mathbf{x}^*) - r\gamma^t \, \left\|\nabla F(\mathbf{x}^t)\right\|^2 + \frac{rM(\gamma^t)^2 K}{2}. \tag{30}$$

We proceed to find a lower bound for the gradient norm $\|\nabla F(\mathbf{x}^t)\|$ in terms of the objective value error $F(\mathbf{x}^t) - F(\mathbf{x}^*)$. Assumption 1 states that the average objective function F is strongly convex with constant $m > 0$. Therefore, for any $\mathbf{y}, \mathbf{z} \in \mathbb{R}^p$ we can write

$$F(\mathbf{y}) \geq F(\mathbf{z}) + \nabla F(\mathbf{z})^T (\mathbf{y} - \mathbf{z}) + \frac{m}{2} \|\mathbf{y} - \mathbf{z}\|^2. \tag{31}$$

For fixed $\mathbf{z}$, the right hand side of (31) is a quadratic function of $\mathbf{y}$ whose minimum argument we can find by setting its gradient to zero. Doing this yields the minimizing argument $\hat{\mathbf{y}} = \mathbf{z} - (1/m)\nabla F(\mathbf{z})$ implying that for all $\mathbf{y}$ we must have

$$\begin{aligned} F(\mathbf{y}) \geq\ & F(\mathbf{w}) + \nabla F(\mathbf{z})^T(\hat{\mathbf{y}} - \mathbf{z}) + \frac{m}{2}\|\hat{\mathbf{y}} - \mathbf{z}\|^2 \\ =\ & F(\mathbf{z}) - \frac{1}{2m}\|\nabla F(\mathbf{z})\|^2. \end{aligned} \tag{32}$$

Observe that the bound in (32) holds true for all $\mathbf{y}$ and $\mathbf{z}$. Setting values $\mathbf{y} = \mathbf{x}^*$ and $\mathbf{z} = \mathbf{x}^t$ in (32) and rearranging the terms yields a lower bound for the squared gradient norm $\|\nabla F(\mathbf{x}^t)\|^2$ as

$$\|\nabla F(\mathbf{x}^t)\|^2 \geq 2m(F(\mathbf{x}^t) - F(\mathbf{x}^*)) \tag{33}$$

Substituting the lower bound in (33) by the norm of gradient square $\|\nabla F(\mathbf{x}^t)\|^2$ in (30) follows the claim in (10).

C Proof of Theorem 1

We use the relationship in (10) to build a supermartingale sequence. To do so, define the stochastic process α^t as

$$\alpha^t := F(\mathbf{x}^t) - F(\mathbf{x}^*) + \frac{rMK}{2}\sum_{u=t}^{\infty}(\gamma^u)^2. \tag{34}$$

Note that α^t is well-defined because $\sum_{u=t}^{\infty}(\gamma^u)^2 \leq \sum_{u=0}^{\infty}(\gamma^u)^2 < \infty$ is summable. Further define the sequence β_t with values

$$\beta^t :=\ 2m\gamma^t r(F(\mathbf{x}^t) - F(\mathbf{x}^*)). \tag{35}$$

The definitions of sequences α^t and β^t in (34) and (35), respectively, and the inequality in (10) imply that the expected value α^{t+1} given $\mathcal{F}^t$ can be written as

$$\mathbb{E}\left[\alpha^{t+1} \,\middle|\, \mathcal{F}^t\right] \ \leq\ \alpha^t - \beta^t. \tag{36}$$

Since the sequences α^t and β^t are nonnegative it follows from (36) that they satisfy the conditions of the supermartingale convergence theorem. Therefore, we obtain that: (i) The sequence α^t converges almost surely to a limit. (ii) The sum $\sum_{t=0}^{\infty}\beta^t < \infty$ is almost surely finite. The latter result yields

$$\sum_{t=0}^{\infty} 2m\gamma^t r(F(\mathbf{x}^t) - F(\mathbf{x}^*)) < \infty. \qquad \text{a.s.} \tag{37}$$

Since the sequence of step sizes is non-summable there exits a subsequence of sequence $F(\mathbf{x}^t) - F(\mathbf{x}^*)$ which is converging to null. This observation is equivalent to almost sure convergence of $\liminf F(\mathbf{x}^t) - F(\mathbf{x}^*)$ to null

$$\liminf_{t\to\infty} F(\mathbf{x}^t) - F(\mathbf{x}^*) = 0. \qquad \text{a.s.} \tag{38}$$

Based on the martingale convergence theorem for the sequences α^t and β^t in relation (36), the sequence α^t almost surely converges to a limit. Consider the definition of α^t in (34). Observe that the sum $\sum_{u=t}^{\infty}(\gamma^u)^2$ is deterministic and its limit is null. Therefore, the sequence of the objective function value error $F(\mathbf{x}^t) - F(\mathbf{x}^*)$ almost surely converges to a limit. This observation in association with the result in (38) implies that the whole sequence of $F(\mathbf{x}^t) - F(\mathbf{x}^*)$ converges almost surely to null,

$$\lim_{t\to\infty}\ F(\mathbf{x}^t) - F(\mathbf{x}^*) = 0. \qquad \text{a.s.} \tag{39}$$

The last step is to prove almost sure convergence of the sequence $\|\mathbf{x}^t - \mathbf{x}^*\|^2$ to null, as a result of the limit in (39). To do so, we follow by proving a lower bound for the objective function value error $F(\mathbf{x}^t) - F(\mathbf{x}^*)$ in terms of the squared norm error $\|\mathbf{x}^t - \mathbf{x}^*\|^2$. According to the strong convexity assumption, we can write the following inequality

$$F(\mathbf{x}^t) \geq F(\mathbf{x}^*) + \nabla F(\mathbf{x}^*)^T(\mathbf{x}^t - \mathbf{x}^*) + \frac{m}{2}\|\mathbf{x}^t - \mathbf{x}^*\|^2 \tag{40}$$

Observe that the gradient of the optimal point is the null vector, i.e., $\nabla F(\mathbf{x}^*) = \mathbf{0}$. This observation and rearranging the terms in (40) imply that

$$F(\mathbf{x}^t) - F(\mathbf{x}^*) \geq \frac{m}{2}\|\mathbf{x}^t - \mathbf{x}^*\|^2. \tag{41}$$

The upper bound in (41) for the squared norm $\|\mathbf{x}^t - \mathbf{x}^*\|^2$ in association with the fact that the sequence $F(\mathbf{x}^t) - F(\mathbf{x}^*)$ almost surely converges to null, leads to the conclusion that the sequence $\|\mathbf{x}^t - \mathbf{x}^*\|^2$ almost surely converges to zero. Hence, the claim in (11) is valid.

The next step is to study the convergence rate of RAPSA in expectation. In this step we assume that the diminishing stepsize is defined as $\gamma^t = \gamma^0 T^0/(t + T^0)$. Recall the inequality in (10). Substitute γ^t by $\gamma^0 T^0/(t + T^0)$ and compute the expected value of (10) given $\mathcal{F}^0$ to obtain

$$\mathbb{E}\left[F(\mathbf{x}^{t+1}) - F(\mathbf{x}^*)\right] \leq \left(1 - \frac{2mr\gamma^0 T^0}{(t + T^0)}\right) \mathbb{E}\left[F(\mathbf{x}^t) - F(\mathbf{x}^*)\right] + \frac{rMK(\gamma^0 T^0)^2}{2(t + T^0)^2}. \tag{42}$$

We use the following lemma to show that the result in (42) implies sublinear convergence of the sequence of expected objective value error $\mathbb{E}\left[F(\mathbf{x}^t) - F(\mathbf{x}^*)\right]$.

Lemma 2. *Let $c > 1$, $b > 0$ and $t^0 > 0$ be given constants and $u_t \geq 0$ be a nonnegative sequence that satisfies the inequality*

$$u^{t+1} \leq \left(1 - \frac{c}{t + t^0}\right) u^t + \frac{b}{(t + t^0)^2}, \tag{43}$$

for all times $t \geq 0$. The sequence u^t is then bounded as

$$u^t \leq \frac{Q}{t + t^0}, \tag{44}$$

for all times $t \geq 0$, where the constant Q is defined as $Q := \max\{b/(c-1),\ t^0 u^0\}$.

Proof: See ..... ■

Lemma 2 shows that if a sequence u^t satisfies the condition in (43) then the sequence u^t converges to null at least with the rate of $O(1/t)$. By assigning values $t^0 = T^0$, $u^t = \mathbb{E}\left[F(\mathbf{x}^t) - F(\mathbf{x}^*)\right]$, $c = 2mr\gamma^0 T^0$, and $b = rMK(\gamma^0 T^0)^2/2$, the relation in (42) implies that the inequality in (43) is satisfied for the case that $2mr\gamma^0 T^0 > 1$. Therefore, the result in (44) holds and we can conclude that

$$\mathbb{E}\left[F(\mathbf{x}^t) - F(\mathbf{x}^*)\right] \leq \frac{C}{t + T^0}, \tag{45}$$

where the constant C is defined as

$$C = \max\left\{\frac{rMK(\gamma^0 T^0)^2}{4rm\gamma^0 T^0 - 2},\ T^0(F(\mathbf{x}^0) - F(\mathbf{x}^*))\right\} \tag{46}$$

D Proof of Theorem 2

To prove the claim in (14) we use the relationship in (10) to construct a supermartingale. Define the stochastic process α^t with values

$$\alpha^t = \big(F(\mathbf{x}^t) - F(\mathbf{x}^*)\big) \times \mathbf{1}\left(\min_{u \leq t}\ F(\mathbf{x}^u) - F(\mathbf{x}^*) > \frac{\gamma MK}{4m}\right) \tag{47}$$

The process α^t tracks the optimality gap $F(\mathbf{x}^t) - F(\mathbf{x}^*)$ until the gap becomes smaller than $\gamma MK/2m$ for the first time at which point it becomes $\alpha^t = 0$. Notice that the stochastic process α^t is always non-negative, i.e., $\alpha^t \geq 0$. Likewise, we define the stochastic process β^t as

$$\beta^t = 2\gamma mr\left(F(\mathbf{x}^t) - F(\mathbf{x}^*) - \frac{\gamma MK}{4m}\right) \times \mathbf{1}\left(\min_{u \leq t}\ F(\mathbf{x}^u) - F(\mathbf{x}^*) > \frac{\gamma MK}{4m}\right), \tag{48}$$

which follows $2\gamma mr\left(F(\mathbf{x}^t) - F(\mathbf{x}^*) - \gamma MK/4m\right)$ until the time that the optimality gap $F(\mathbf{x}^t) - F(\mathbf{x}^*)$ becomes smaller than $\gamma MK/2m$ for the first time. After this moment the stochastic process

β^t becomes null. According to the definition of β^t in (48), the stochastic process satisfies $\beta^t \geq 0$ for all $t \geq 0$. Based on the relationship (10) and the definitions of stochastic processes α^t and β^t in (47) and (48) we obtain that for all times $t \geq 0$

$$\mathbb{E}\left[\alpha^{t+1} \mid \mathcal{F}^t\right] \leq \alpha^t - \beta^t. \tag{49}$$

To check the validity of (49) we first consider the case that $\min_{u \leq t} F(\mathbf{x}^u) - F(\mathbf{x}^*) > \gamma MK/4m$ holds. In this scenario we can simply the stochastic processes in (47) and (48) as $\alpha^t = F(\mathbf{x}^t) - F(\mathbf{x}^*)$ and $\beta^t = 2\gamma mr\left(F(\mathbf{x}^t) - F(\mathbf{x}^*) - \gamma MK/4m\right)$. Therefore, according to the inequality in (10) the result in (49) is valid. The second scenario that we check is $\min_{u \leq t} F(\mathbf{x}^u) - F(\mathbf{x}^*) \leq \gamma MK/4m$. Based on the definitions of stochastic processes α^t and β^t, both of these two sequences are equal to 0. Further, notice that when $\alpha^t = 0$, it follows that $\alpha^{t+1} = 0$. Hence, the relationship in (49) is true.

Given the relation in (49) and non-negativity of stochastic processes α^t and β^t we obtain that α^t is a supermartingale. The supermartingale convergence theorem yields: i) The sequence α^t converges to a limit almost surely. ii) The sum $\sum_{t=1}^{\infty} \beta^t$ is finite almost surely. The latter result implies that the sequence β^t is converging to null almost surely. I.e.,

$$\lim_{t \to \infty} \beta^t \;=\; 0 \quad a.s. \tag{50}$$

Based on the definition of β^t in (48), the limit in (50) is true if one of the following events holds: i) The indicator function is null after for large t. ii) The limit $\lim_{t \to \infty} \left(F(\mathbf{x}^t) - F(\mathbf{x}^*) - \gamma MK/4m\right) = 0$ holds true. From any of these two events we it is implied that

$$\liminf_{t \to \infty} \; F(\mathbf{x}^t) - F(\mathbf{x}^*) \;\leq\; \frac{\gamma MK}{4m} \quad a.s. \tag{51}$$

Therefore, the claim in (14) is valid. The result in (51) shows the objective function value sequence $F(\mathbf{x}^t)$ almost sure converges to a neighborhood of the optimal objective function value $F(\mathbf{x}^*)$.

We proceed to prove the result in (15). Compute the expected value of (10) given $\mathcal{F}^0$ and set $\gamma^t = \gamma$ to obtain

$$\mathbb{E}\left[F(\mathbf{x}^{t+1}) - F(\mathbf{x}^*)\right] \leq (1 - 2m\gamma r)\, \mathbb{E}\left[F(\mathbf{x}^t) - F(\mathbf{x}^*)\right] + \frac{rMK\gamma^2}{2}. \tag{52}$$

Notice that the expression in (52) provides an upper bound for the expected value of objective function error $\mathbb{E}\left[F(\mathbf{x}^{t+1}) - F(\mathbf{x}^*)\right]$ in terms of its previous value $\mathbb{E}\left[F(\mathbf{x}^t) - F(\mathbf{x}^*)\right]$ and an error term. Rewriting the relation in (52) for step $t-1$ leads to

$$\mathbb{E}\left[F(\mathbf{x}^t) - F(\mathbf{x}^*)\right] \leq (1 - 2m\gamma r)\, \mathbb{E}\left[F(\mathbf{x}^{t-1}) - F(\mathbf{x}^*)\right] + \frac{rMK\gamma^2}{2}. \tag{53}$$

Substituting the upper bound in (53) for the expectation $\mathbb{E}\left[F(\mathbf{x}^t) - F(\mathbf{x}^*)\right]$ in (52) follows an upper bound for the expected error $\mathbb{E}\left[F(\mathbf{x}^{t+1}) - F(\mathbf{x}^*)\right]$ as

$$\mathbb{E}\left[F(\mathbf{x}^{t+1}) - F(\mathbf{x}^*)\right] \leq (1 - 2m\gamma r)^2\, \mathbb{E}\left[F(\mathbf{x}^{t-1}) - F(\mathbf{x}^*)\right] + \frac{rMK\gamma^2}{2}\left(1 + (1 - 2mr\gamma)\right). \tag{54}$$

By recursively applying the steps in (53) and (54) we can bound the expected objective function error $\mathbb{E}\left[F(\mathbf{x}^{t+1}) - F(\mathbf{x}^*)\right]$ in terms of the initial objective function error $F(\mathbf{x}^0) - F(\mathbf{x}^*)$ and the accumulation of the errors as

$$\mathbb{E}\left[F(\mathbf{x}^{t+1}) - F(\mathbf{x}^*)\right] \leq (1 - 2m\gamma r)^{t+1}\left(F(\mathbf{x}^0) - F(\mathbf{x}^*)\right) + \frac{rMK\gamma^2}{2}\sum_{u=0}^{t}(1 - 2mr\gamma)^u. \tag{55}$$

Substituting t by $t-1$ and simplifying the sum in the right hand side of (55) yields

$$\mathbb{E}\left[F(\mathbf{x}^t) - F(\mathbf{x}^*)\right] \leq (1 - 2m\gamma r)^t\left(F(\mathbf{x}^0) - F(\mathbf{x}^*)\right) + \frac{MK\gamma}{4m}\left[1 - (1 - 2mr\gamma)^t\right]. \tag{56}$$

Observing that the term $1 - (1 - 2mr\gamma)^t$ in the right hand side of (56) is strictly smaller than 1 for the stepsize $\gamma < 1/(2mr)$, the claim in (15) follows.

E Implementation of ARAPS

For reference, ARAPSA is also summarized in algorithmic form in Algorithm 2. Steps 2 and 3 are devoted to assigning random blocks to the processors. In Step 2 a subset of available blocks $\mathcal{I}^t$ is chosen. These blocks are assigned to different processors in Step 3. In Step 5 processors compute the partial stochastic gradient corresponding to their assigned blocks $\nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \mathbf{\Theta}_i^t)$ using the acquired samples in Step 4. Steps 6 and 7 are devoted to the computation of the ARAPSA descent direction $\hat{\mathbf{d}}_i^t$. In Step 6 the approximate Hessian inverse $\hat{\mathbf{B}}_i^{t,0}$ for block $\mathbf{x}_i$ is initialized as $\hat{\mathbf{B}}_i^{t,0} = \eta_i^t \mathbf{I}$ which is a scaled identity matrix using the expression for η_i^t in (20) for $t > 0$. The initial value of η_i^t is $\eta_i^0 = 1$. In Step 7 we use Algorithm 1 for efficient computation of the descent direction $\hat{\mathbf{d}}_i^t = \hat{\mathbf{B}}_i^t \nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \mathbf{\Theta}_i^t)$. The descent direction $\hat{\mathbf{d}}_i^t$ is used to update the block $\mathbf{x}_i^t$ with stepsize γ^t in Step 8. Step 9 determines the value of the partial stochastic gradient $\nabla_{\mathbf{x}_i} f(\mathbf{x}^{t+1}, \mathbf{\Theta}_i^t)$ which is required for the computation of stochastic gradient variation $\hat{\mathbf{r}}_i^t$. In Step 10 the variable variation $\mathbf{v}_i^t$ and stochastic gradient variation $\hat{\mathbf{r}}_i^t$ associated with block $\mathbf{x}_i$ are computed to be used in the next iteration.

Algorithm 1 Computation of the ARAPSA step $\hat{\mathbf{d}}_i^t = \hat{\mathbf{B}}_i^t \nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \mathbf{\Theta}_i^t)$ for block $\mathbf{x}_i$.

1: **function** $\hat{\mathbf{d}}_i^t = \mathbf{q}^\tau$ = ARAPSA Step$\Big(\hat{\mathbf{B}}_i^{t,0},\ \mathbf{p}^0 = \nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \mathbf{\Theta}_i^t),\ \{\mathbf{v}_i^u, \hat{\mathbf{r}}_i^u\}_{u=t-\tau}^{t-1}\Big)$
2: **for** $u = 0, 1, \ldots, \tau - 1$ **do** {Loop to compute constants α^u and sequence $\mathbf{p}^u$}
3: Compute and store scalar $\alpha^u = \hat{\rho}_i^{t-u-1} (\mathbf{v}_i^{t-u-1})^T \mathbf{p}^u$
4: Update sequence vector $\mathbf{p}^{u+1} = \mathbf{p}^u - \alpha^u \hat{\mathbf{r}}_i^{t-u-1}$.
5: **end for**
6: Multiply $\mathbf{p}^\tau$ by initial matrix: $\mathbf{q}^0 = \hat{\mathbf{B}}_i^{t,0} \mathbf{p}^\tau$
7: **for** $u = 0, 1, \ldots, \tau - 1$ **do** {Loop to compute constants β_u and sequence $\mathbf{q}_u$}
8: Compute scalar $\beta^u = \hat{\rho}_i^{t-\tau+u} (\hat{\mathbf{r}}_i^{t-\tau+u})^T \mathbf{q}^u$
9: Update sequence vector $\mathbf{q}^{u+1} = \mathbf{q}^u + (\alpha^{\tau-u-1} - \beta^u) \mathbf{v}_i^{t-\tau+u}$
10: **end for** {return $\hat{\mathbf{d}}_i^t = \mathbf{q}^\tau$}

Algorithm 2 Accelerated Random Parallel Stochastic Algorithm (ARAPSA) for individual processors

1: **for** $t = 0, 1, 2, \ldots$ **do**
2: Choose uniformly at random set $\mathcal{I}^t \subset \{1, \ldots, B\}$ of block variables to update
3: Assign block variables $\mathcal{S}^t$ to processors in any manner.
4: Choose a set of realizations $\mathbf{\Theta}_i^t$ for the block $\mathbf{x}_i$
5: Compute stochastic gradient : $\nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \mathbf{\Theta}_i^t) = \dfrac{1}{L} \displaystyle\sum_{\boldsymbol{\theta} \in \mathbf{\Theta}_i^t} \nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \boldsymbol{\theta})$ [cf. (3)]
6: Compute the initial Hessian inverse approximation: $\hat{\mathbf{B}}_i^{t,0} = \eta_i^t \mathbf{I}$
7: Compute descent direction: $\hat{\mathbf{d}}_i^t =$ oLBFGS Step $\Big(\hat{\mathbf{B}}_i^{t,0},\ \nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \mathbf{\Theta}_i^t),\ \{\mathbf{v}_i^u, \hat{\mathbf{r}}_i^u\}_{u=t-\tau}^{t-1}\Big)$
8: Update the coordinates of the decision variable $\mathbf{x}_i^{t+1} = \mathbf{x}_i^t - \gamma^t \hat{\mathbf{d}}_i^t$
9: Compute *updated* stochastic gradient: $\nabla_{\mathbf{x}_i} f(\mathbf{x}^{t+1}, \mathbf{\Theta}_i^t) = \dfrac{1}{L} \displaystyle\sum_{\boldsymbol{\theta} \in \mathbf{\Theta}_i^t} \nabla_{\mathbf{x}_i} f(\mathbf{x}^{t+1}, \boldsymbol{\theta})$ [cf. (3)]
10: Update variations $\mathbf{v}_i^t = \mathbf{x}_i^{t+1} - \mathbf{x}_i^t$ and $\hat{\mathbf{r}}_i^t = \nabla_{\mathbf{x}_i} f(\mathbf{x}^{t+1}, \mathbf{\Theta}_i^t) - \nabla_{\mathbf{x}_i} f(\mathbf{x}^t, \mathbf{\Theta}_i^t)$ [cf.(19)]
11: **end for**