**Our Contributions**

- Propose a novel doubly stochastic, parallel and asynchronous algorithm which can be interpreted as stochastic block coordinate stochastic gradient method
- Analyze the algorithm and provide convergence results with precise constants
- Characterize speedup explicitly under a reasonable model for computation time

**Problem Statement**

Consider the problem

\[
\min_{x \in \mathbb{R}^d} f(x)
\]

where \(f : \mathbb{R}^d \to \mathbb{R}\) is a \(m\)-strongly convex and has the form

\[
f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x).
\]

**Goal:** Design a parallel and asynchronous algorithm, with read and writes to shared memory

**Parallel:** Computation done over multiple cores

**Asynchronous:** Each core runs on its own clock

**Shared memory:** Single memory is used by cores to communicate their updates

**Assumption 1.** Instantaneous read and writes (can be relaxed using techniques in [1] with minor changes in the prefactor of our main result)

**Assumption 2.** Between a core reading and writing there are at most \(n\) iterations (can be relaxed using techniques from stochastic processes)

**Assumption 3.** \(\|\nabla f_i(x)\| \leq M\) for all \(i\) and \(x\) (can be relaxed using techniques in [3])

**Related Work**

- [1, 2] discuss an asynchronous and parallel stochastic gradient method. They assume that \(f_i\)'s depend on \(x \in \mathbb{R}^d\) in a sparse fashion, and their results involve factors relating to these dependencies. **Our work does not assume such a sparse dependence.**
- [3] propose an asynchronous and parallel stochastic coordinate scheme for a broad class of convex functions, where each core updates a single random coordinate at each iteration using exact gradient. So each update can be assumed to take \(O(n)\) time in our function model (against \(O(n^d)\) time to compute the full gradient). While in our scheme each update takes \(O(pd)\) where \(p \in (0, 1]\

**RAHAN Algorithm**

**parameters:** step size \(\alpha\), time horizon \(T\), sparsification factor \(p\)

On each core:

**while total number of writes to shared memory \(\leq T\) do**

// read from the shared memory

\(x \leftarrow \text{current iterate from shared memory}\)

// stochastic gradient step

\(i \leftarrow \text{uniform}\{1, \ldots, n\}\)

// stochastic block coordinate step

\(S \leftarrow \text{each coordinate independently with probability } p\)

// write to the shared memory

\(x \leftarrow x - \alpha P_S \nabla f_i(x)\)

end

where \(P_S\) is the projection operator onto set \(S\).

**Main Results**

**Theorem 1.** For our algorithm, choose target error \(\epsilon\) and a constant step size \(\alpha\) as,

\[
\epsilon \leq \frac{pM^2(1 + 2pr)}{4m^2r}, \quad \alpha = \frac{em}{M^2(1 + 2pr)}
\]

then \(\mathbb{E}[\|x_T - x_0\|^2] \leq \epsilon\) for

\[
T \geq \left(1 + \frac{2pr}{p}\right) \left(\frac{2M^2}{em^2} \log\left(\frac{2m}{\epsilon}\right)\right).
\]

**Remark:** Serial SGD results are recovered by setting \(\tau = 0, p = 1\).

**Corollary 1.** If \(K\) cores are used, and each core’s computation time for \(p\) fraction of gradient coordinates is \(t_0 + pg\), then for optimal sparsification factor \(p = \sqrt{\gamma/(2\gamma)}\) where \(\gamma = t_0/g\), the speedup \(\tau\) wrt the serial SGD in wall time is

\[
K\sqrt{\frac{e^{\gamma}}{(1 + e^{\gamma})}}.
\]

Further if \(\tau = \beta K\), then the speedup for \(K = 1/(\beta\gamma)\) is \((K + 1)/4\).

**Discussion of Results**

- Regarding assumptions in Corollary 1:
  - The results can be extended to any computation time model of the form \(t_0 + p\gamma\) for any given \(\alpha > 0\)
  - Number of clashes scales linearly with the number of cores which is consistent with queueing models
- Instantaneous read and writes are not necessary for the analysis and we can get rid of it using techniques from [1]; the prefactor in Theorem 1 is affected mildly
- The scheme as presented is a finite time horizon scheme, but it can be adapted to any-time horizon by reducing step size with time (as in epoch SGD)

**Key Ideas in the Proof**

Let \(a_k = \mathbb{E}[\|x_k - x_*\|^2]\) and \(\hat{x}_k\) be the iterate used to compute the gradient for the \(k\)th write, strong convexity leads to the recursion equation

\[
a_{k+1} \leq (1 - \alpha pm)a_k + \alpha^2 \mathbb{E}[\|g_k(\hat{x}_k)\|^2] + 2\alpha \mathbb{E}[\|\hat{x}_k - x_k, g_k(\hat{x}_k)\|].
\]

Bound the three terms as below:

- \(A\) can be controlled by bounded gradient assumption
- \(B\) and \(C\) can be controlled by bounded gradients and finite number of overwrites incorporating the sparsification factor

The rest of the proof relies on independent sampling of functions and coordinates, and careful bounding of resultant terms using algebraic techniques.

**Future Research**

- Derive bounds for the function values following the analysis in [3] for our algorithm
- Analyze relations to ‘asynchronous stochastic approximation’

**References**

