Doubly Sparse Asynchronous Learning for Stochastic Composite Optimization

Runxue Bao, Xidong Wu, Wenhan Xian, Heng Huang

Electrical and Computer Engineering, University of Pittsburgh, PA 15213, United States

IJCAI-ECAI 2022

Background

- We consider the composite optimization problem involving a data fitting function $\mathcal{F}(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(a_i^\top x)$ plus a block-separable regularizer $\Omega(x) = \sum_{j=1}^{q} \Omega_j(x_{\mathcal{G}_j})$ as:

$$\min_{x \in \Re^p} \mathcal{P}(x) := \mathcal{F}(x) + \lambda \Omega(x). \tag{1}$$

  where $A = [a_1, \cdots, a_n]^\top \in \Re^{n \times p}$ is the design matrix, $\mathcal{G}$ is the partition, λ is the regularization parameter and $x \in \Re^p$ is the model coefficients.
- Most existing parallel learning methods focus on improving the algorithm efficiency in terms of sample complexity and thus suffer from high computation costs and memory burden in the high-dimensional setting.

Objective: To accelerate high-dimensional models by simultaneously enjoying the model sparsity and data sparsity.

Proposed Method

Algorithm 1 Sha-DSAL

```
Input: $x^0_{\mathcal{B}_0} \in \Re^p$, step size $\eta$, inner loops $K$.
for $s = 0$ to $S - 1$ do
    All threads parallelly compute $\nabla \mathcal{F}(x^0_{\mathcal{B}_s})$.
    Compute dual variable $y^s$ and update $\mathcal{B}_{s+1}$ from $\mathcal{B}_s$ by (2).
    Update $A_{\mathcal{B}_{s+1}}, x^0_{\mathcal{B}_{s+1}}, \nabla \mathcal{F}(x^0_{\mathcal{B}_{s+1}})$
    For each thread, do:
    for $t = 0$ to $K - 1$ do
        Read $\hat{x}^t_{\mathcal{B}_{s+1}}$ from the shared memory.
        Randomly sample $i$ from $\{1, 2, \ldots, n\}$.
        Compute $v^s_t$ by (3).
        $\delta^s_t = \mathrm{prox}_{\eta\lambda\phi_i}(\hat{x}^t_{\mathcal{B}_{s+1}} - \eta v^s_t) - \hat{x}^t_{\mathcal{B}_{s+1}}$.
        $x^{t+1}_{\mathcal{B}_{s+1}} = x^t_{\mathcal{B}_{s+1}} + \delta^s_t$.
    end for
    $x_{\mathcal{B}_{s+1}} = x^K_{\mathcal{B}_{s+1}}, x^0_{\mathcal{B}_{s+1}} = x_{\mathcal{B}_{s+1}}$.
end for
```

Proposed Method

Eliminating Step: We update $\mathcal{B}_{s+1}$ for $\forall j \in \mathcal{B}_s$ as:

$$\Omega_j^D(A_j^\top y^s) + \Omega_j^D(A_j)\sqrt{2L(\mathcal{P}(x^0_{\mathcal{B}_s}) - D(y^s))} \geq n\lambda. \quad (2)$$

Variance-Reduced Sparse Gradient: Define Ψ_i as the set of blocks that intersect the nonzero coefficients of ∇f_i, let $n_{\mathcal{G}}$ be the number of occurrences that $\mathcal{G} \in \Psi_i$, if $n_{\mathcal{G}} > 0$, we define $d_{\mathcal{G}} = n/n_{\mathcal{G}}$. Thus, define diagonal matrix for block i as $[D_i]_{\mathcal{G},\mathcal{G}} = d_{\mathcal{G}} I_{|\mathcal{G}|}$, the gradient over $\mathcal{B}_{s+1}$ can be computed as

$$v_t^s = \nabla f_i(a_{i,\mathcal{B}_{s+1}}^\top \hat{x}^t_{\mathcal{B}_{s+1}}) - \nabla f_i(a_{i,\mathcal{B}_{s+1}}^\top x^0_{\mathcal{B}_{s+1}}) + D_{i,\mathcal{B}_{s+1}} \nabla \mathcal{F}(x^0_{\mathcal{B}_{s+1}}). \quad (3)$$

Sparse Proximal Gradient Update: Define $\phi_i(x) = \sum_{\mathcal{G}\in\Psi_i} d_{\mathcal{G}} \Omega_{\mathcal{G}}(x)$, the new proximal operator can be computed as

$$\mathrm{prox}_{\eta\lambda\phi_i}(x') = \arg\min_x \frac{1}{2\eta}\|x - x'\|^2 + \lambda\phi_i(x). \quad (4)$$

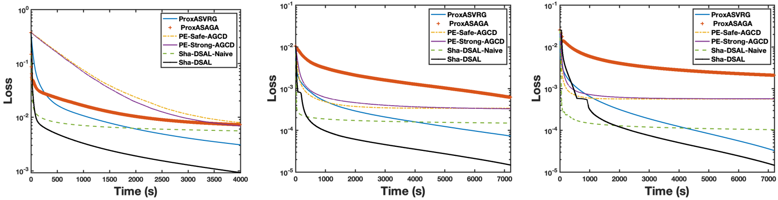
Theoretical Analysis

Linear Convergence: Suppose $\tau \leq \frac{1}{10\sqrt{\Delta}}$, let step size $\eta = \min\{\frac{1}{24\kappa L}, \frac{\kappa}{2L}, \frac{\kappa}{10\tau L}\}$, inner loop size $K = \frac{4\log 3}{\eta\mu}$, we have

$$\mathbb{E}\left\|x_{\mathcal{B}_S} - x^*_{\mathcal{B}_S}\right\|^2 \leq (2/3)^S \left\|x_0 - x^*\right\|^2. \tag{5}$$

Elimination Ability: Equicorrelation set [4] is defined as $\mathcal{B}^* := \{j \in \{1, 2, \ldots, q\} : \Omega_j^D(A_j^\top y^*) = n\lambda\}$. As DSAL converges, there exists an iteration number $S_0 \in \mathbb{N}$, s.t. $\forall s \geq S_0$, any variable block $j \notin \mathcal{B}^*$ is eliminated by DSAL almost surely.

Convergence Results

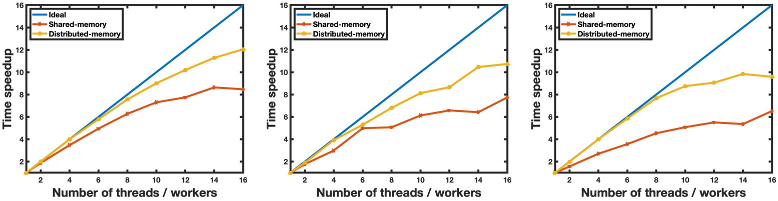


(a) KDD2010 (b) Avazu-app (c) Avazu-site

Figure: Convergence results on shared-memory architecture with 8 threads.

We compare six asynchronous methods: 1) PE-Strong-AGCD: parallel strong elimination in [1]; 2) PE-Safe-AGCD: parallel static safe elimination in [1]; 3) ProxASAGA [3]; 4) ProxASVRG [2]; 5) Sha-DSAL-Naive; 6) Our Sha-DSAL.

Linear Speedup Property



(a) KDD2010 (b) Avazu-app (c) Avazu-site

Figure: Convergence results with different number of threads/workers.

Conclusion

- We propose a novel accelerated doubly sparse asynchronous learning method for stochastic composite optimization and apply it on shared-memory and distributed-memory architecture respectively.
- DSAL can simultaneously enjoy the model sparsity and data sparsity.
- We rigorously prove DSAL can achieve a linear convergence rate, reduce the per-iteration cost, and achieve a lower overall computational complexity under the strongly convex condition.
- We empirically show that DSAL can simultaneously achieve significant acceleration and linear speedup property.

Thank You!

Q. Li, S. Qiu, S. Ji, P. M. Thompson, J. Ye, and J. Wang.
Parallel lasso screening for big data optimization.
In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1705–1714, 2016.

Q. Meng, W. Chen, J. Yu, T. Wang, Z.-M. Ma, and T.-Y. Liu.
Asynchronous stochastic proximal optimization algorithms with variance reduction.
In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

F. Pedregosa, R. Leblond, and S. Lacoste-Julien.
Breaking the nonsmooth barrier: A scalable parallel method for composite optimization.
In *NIPS*, 2017.

R. J. Tibshirani et al.
The lasso problem and uniqueness.
Electronic Journal of statistics, 7:1456–1490, 2013.