Network (In)dependence in Distributed Optimization

Artin Spiridonoff, Alex Olshevsky and Ioannis Ch. Paschalidis

Department of Electrical and Computer Engineering and Division of Systems Engineering, Boston University

BOSTON UNIVERSITY

Abstract

We consider a harsh network model characterized by asynchronous updates, message delays, unpredictable message losses, and directed communication among nodes. In this setting, we analyze a modification of the Gradient-Push method for distributed optimization. We show that our proposed method asymptotically performs as well as the best bounds on centralized gradient descent [1].

Model

We consider the separable optimization problem:

$$\min_{\mathbf{z}\in\mathbb{R}^d} F(\mathbf{z}) = \sum_{i=1}^{n} f_i(\mathbf{z}),$$

where each f_i is known only to agent i.

Agents communicate through a strongly connected directed graph. We allow for:

- Bounded delays L_d,
- Bounded consecutive link failures L_f,
- Agents waking up at least once every $L_u > 0$ iterations.
- Gradients of each f_i are corrupted by a zero-mean independent random variable ε_i, where $\|\varepsilon_i\| \leq b_i$ and $\mathbb{E}[\|\varepsilon_i\|^2] \leq \sigma_i^2$.
- Each $f_i(\mathbf{z})$ is μ_i-strongly convex over $\mathbb{R}^d$.
- The gradient of each $f_i(\mathbf{z})$ is L-Lipschitz continuous.

We propose the following Robust Asynchronous Stochastic Gradient-Push (RASGP) algorithm for the above scenario.

Performance Guarantee

Theorem 1. *Suppose assumptions above hold, then the RASGP algorithm with the step-size $\alpha(k) = n/(\mu k)$ for $k \geq 1$, will converge to the unique optimum $\mathbf{z}^*$ with the following asymptotic rate:*

$$\mathbb{E}\left[\|\mathbf{z}_i(k) - \mathbf{z}^*\|^2\right] \leq \frac{L_u \sigma^2}{k\mu^2} + O_k\left(\frac{1}{k^{1.5}}\right), \qquad \forall i,$$

where $\sigma^2 := \sum_i \sigma_i^2$ and $\mu := \sum_i \mu_i$.

The leading term matches the best bounds for (centralized) gradient descent that takes steps in the direction of the sum of the noisy gradients of $f_1(\mathbf{z}), \ldots, f_n(\mathbf{z})$ (see [2]), when $L_u = 1$.

Main Algorithm

The RASGP is based on a standard idea of mixing consensus and gradient steps, first analyzed in [3]. The push-sum scheme, inspired by [4], is used instead of the consensus scheme, which allows us to handle delays, and message losses. We note that a new step-size strategy is used to handle asynchronicity.

Algorithm 1 Robust Asynchronous Stochastic Gradient-Push (RASGP)

```
Initialize the algorithm with y(0) = 1, φ_i^x(0) = 0, φ_i^y(0) = 0, κ_i(0) = −1, ∀i ∈ {1,...,n} and ρ_ij^x(0) = 0, ρ_ij^y(0) = 0, κ_ij(0) = 0, ∀(j,i) ∈ E.
At every iteration k = 0,1,2,···, for every node i:
if node i wakes up then
    β_i(k) = Σ_{t=κ_i+1}^{k} α(t);                                  (to handle asynchrony)
    x_i ← x_i − β_i(k) ĝ_i(k);                         (ĝ_i(k) = ∇f_i(z_i(k)) + ε_i(k))
    κ_i ← k;
    φ_i^x ← φ_i^x + x_i/(d_i^+ + 1), φ_i^y ← φ_i^y + y_i/(d_i^+ + 1);   (to handle delays and link failures)
    x_i ← x_i/(d_i^+ + 1), y_i ← y_i/(d_i^+ + 1);
    Node i broadcasts (φ_i^x, φ_i^y, κ_i) to its out-neighbors: N_i^+
    Processing the received messages
    for (φ_j^x, φ_j^y, κ'_j) in the inbox do
        if κ'_j > κ_ij then
            ρ_ij^{*x} ← φ_j^x, ρ_ij^{*y} ← φ_j^y;
            κ_ij ← k'_j;
        end if
    end for
    x_i ← x_i + Σ_{j∈N_i^-} (ρ_ij^{*x} − ρ_ij^x), y_i ← y_i + Σ_{j∈N_i^-} (ρ_ij^{*y} − ρ_ij^y);
    ρ_ij^x ← ρ_ij^{*x}, ρ_ij^y ← ρ_ij^{*y};
    z_i ← x_i/y_i;
end if
Other variables remain unchanged.
```

Numerical Simulations

Binary class, strongly convex, and smooth Support Vector Machine (SVM):

$$F(\boldsymbol{\omega}, \gamma) = \frac{1}{2}\left(\|\boldsymbol{\omega}\|^2 + \gamma^2\right) + C\sum_{j=1}^{N} h(b_j(\mathbf{A}_j^\top \boldsymbol{\omega} + \gamma)),$$

where $(\boldsymbol{\omega}, \gamma) \in \mathbb{R}^d$, $\mathbf{A}_j \in \mathbb{R}^{d-1}, b_j \in \{-1, +1\}$, $j = 1, \ldots, N$, are the data points and their labels, respectively. Here, $h: \mathbb{R} \to \mathbb{R}$ is the smoothed hinge loss. Hence the local objective functions are

$$f_i(\boldsymbol{\omega}_i, \gamma_i) = \frac{1}{2n}\left(\|\boldsymbol{\omega}\|^2 + \gamma^2\right) + C\sum_{j\in D_i} h(b_j(\mathbf{A}_j^\top \boldsymbol{\omega} + \gamma)).$$

Uniform noise of $\varepsilon_i \sim \mathbb{U}[-b/2, b/2]^d$ and $\varepsilon_c \sim \mathbb{U}[-\sqrt{n}b/2, \sqrt{n}b/2]^d$ is added to the gradient estimates of each agent i and the centralized algorithm, respectively. The network is chosen to be a directed cycle graph and a random graph with $n = 50$, $(N = 1000, C = 0.5, d = 3, b = 5$, $L_d \in \{1, 3\}, L_f \in \{1, 3\}, L_u \in \{1, 3\})$.

Remarks

This work presents an algorithm that allows for delays, link failures and asynchrony. Moreover, we showed that the distributed algorithm can asymptotically reach the optimal bound for its centralized counterpart.

References

[1] Alex Olshevsky, Ioannis Ch Paschalidis, and Artin Spiridonoff. Robust asynchronous stochastic gradient-push: asymptotically optimal and network-independent performance for strongly convex functions. *arXiv preprint arXiv:1811.03982*, 2018.

[2] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

[3] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[4] Christoforos N Hadjicostis, Nitin H Vaidya, and Alejandro D Domínguez-García. Robust distributed average consensus via exchange of running sums. *IEEE Transactions on Automatic Control*, 61(6):1492–1507, 2016.