Adaptive Random Walk Gradient Descent for Decentralized Optimization

Tao Sun, Dongsheng Li, and Bao Wang

NUDT & U. of Utah

Table of contents

What is the problem (research area)

The model

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{1, 2, \cdots, n\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ being the set of agents and the set of edges that connect agents, respectively. We consider the **decentralized algorithm** for the minimization problem over graph $\mathcal{G}$

$$\min_{\mathsf{x} \in \mathbb{R}^d} f(\mathsf{x}) := \frac{1}{n} \sum_{i=1}^{n} f_i(\mathsf{x}), \; f_i(\mathsf{x}) = \mathbb{E}_{\xi \sim \mathcal{D}_i} F_i(\mathsf{x}; \xi), \tag{1}$$

where $\mathcal{D}_i$ denotes the data distribution of the i-th client and $F_i(\mathsf{x}; \xi)$ is the loss function associated with the training data ξ.

The algorithm: random walk gradient descent

A token randomly walks over the graph $\mathcal{G}$ to sample the data and updates the parameter.

Random walk gradient descent only involves one edge communication in each iteration, resulting in a minimum communication cost.

Another advantage is that it also applies to the directed graph setting.

What we do

The adaptive random walk

Motivated by the Adam-type algorithms, we propose the adaptive random walk algorithm as follows.

Algorithm 1 Adaptive Random Walk Gradient Descent

Require: parameters $\eta > 0$, $0 \le \theta < 1$, $\delta > 0$

Initialization: $\boldsymbol{g}^0 = \boldsymbol{0}$, $\boldsymbol{m}^0 = \boldsymbol{0}$, $\boldsymbol{v}^0 = \boldsymbol{0}$

for $k = 1, 2, \ldots$

step 1: agent i_k calculates $\boldsymbol{g}^k = \nabla f_{i_k}(\boldsymbol{x}^k)$

step 2: $\boldsymbol{m}^k = \theta \boldsymbol{m}^{k-1} + (1-\theta)\boldsymbol{g}^k$

step 3: $\boldsymbol{v}^k = \boldsymbol{v}^{k-1} + [\boldsymbol{g}^k]^2$

step 4: $\boldsymbol{z}^{k+1} = \boldsymbol{x}^k - \eta \boldsymbol{m}^k / (\boldsymbol{v}^k + \delta \mathbb{I})^{\frac{1}{2}}$

step 5: $\boldsymbol{x}^{k+1} = \arg\min_{\boldsymbol{x} \in \mathcal{K}} \|\boldsymbol{z}^{k+1} - \boldsymbol{x}\|^2_{(\boldsymbol{v}^k + \delta \mathbb{I})^{\frac{1}{2}}}$

step 6: uses random walk to choose a neighbor i_{k+1} and sends $(\boldsymbol{x}^k, \boldsymbol{m}^k, \boldsymbol{v}^k)$ via edge (i_k, i_{k+1}) to i_{k+1}

end for

The convergence

We investigate the adaptive random walk gradient descent and establish its theoretical performance bounds in both convex and nonconvex settings.
In the following, we present the convex one.

Theorem

Let Assumptions 1, 2, 3, 4, and condition

$$\mathbb{E}\|(\mathrm{v}^K+\delta\mathbb{I})^{\frac{1}{2}}\|_1 \leq CK^{\alpha}, \tag{2}$$

hold. Assume $\{\mathrm{x}^k\}_{k\geq 1}$ *is generated by Algorithm 1. By setting* $\eta=\min\{\frac{\ln(1/\sigma(\mathrm{P}))}{\ln(1/\epsilon)},1\}$, *then*

$$\mathbb{E}\left[f\Big(\frac{\sum_{k=1}^{K}\mathrm{x}^k}{K}\Big)-\min f\right]=\mathcal{O}(\epsilon), \tag{3}$$

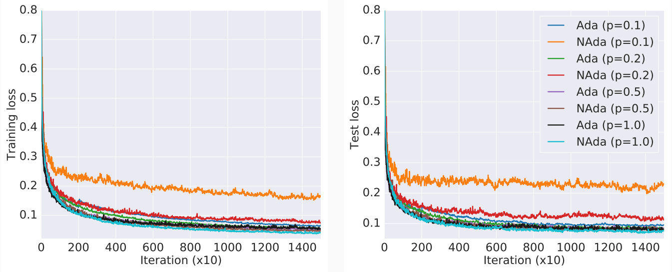
with $K=\widetilde{\mathcal{O}}\Big(\max\Big\{\frac{1}{\epsilon^{\frac{1}{1-\alpha}}[\ln(1/\sigma(\mathrm{P}))]^{\frac{1}{1-\alpha}}},\frac{1}{\epsilon^{\frac{1}{1-\alpha}}}\Big\}\Big)$.

Advantage i

Due to the boundedness of stochastic gradients $\{\mathbf{g}^k\}_{k\geq 0}$, $\alpha = 1/2$ can hold in (2) without any extra assumption.

As the stochastic gradient decay fast (i.e., $\alpha < 1/2$), adaptive random walk gradient descent can be faster than the non-adaptive ones.

What about the new algorithm



Training　　　　Test

Figure 1: Comparison of adaptive and non-adaptive random walk algorithms, with both sparse ($p < 1$) and non-sparse gradients ($p = 1$), for training an MLP model for MNIST classification. In this experiment, we have ten clients connected by a ring graph.

1. Duchi, J. C., Agarwal, A., Johansson, M., and Jordan, M. I. Ergodic mirror descent. SIAM Journal on Optimization, 22(4):1549–1578, 2012.

2. Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12(Jul):2121–2159, 2011.

3. Mao, X., Yuan, K., Hu, Y., Gu, Y., Sayed, A. H., and Yin, W. Walkman: A communication-efficient random-walk algorithm for decentralized optimization. IEEE Transactions on Signal Processing, 68:2513–2528, 2020.

4. Sun, T., Sun, Y., and Yin, W. On markov chain gradient descent. Advances in neural information processing systems, 2018.

5. Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In ICLR (Poster), 2015.