Research Statement

On Theory and Application of Decentralized Optimization

Yao Li
Department of Mathematics
Department of Computational Mathematics, Science and Engineering

Decentralized optimization can date back to Tsitsiklis (1984). The main difference from most of distributed algorithms is that there is no central master to broadcast global information to worker nodes, while only local data will be exchanged between accessible neighbours along edges of the network. It's very similar to the spread of gossip, hence sometimes it's also called gossip algorithm (e.g. Xiao and Boyd, 2004).

The kernel of decentralization can be revealed from an average consensus problem. If each agent privately knows a number, how can we design an algorithm such that everyone can get the average number without exchanging the number with the rest of all agents? The mathematical form is described as the following:

$$\begin{aligned} &\underset{x_i \in \mathbb{R}}{\text{minimize}} \ \|x_1 - b_1\|_2^2 + \cdots + \|x_n - b_n\|_2^2 \\ &\text{subject to } x_1 = x_2 = \cdots = x_n, \end{aligned} \tag{1}$$

where $(i, j) \in G$ forms a fully connected undirected graph G, i.e., their is no agent isolated from the network and the communication is undirectional. The classic gossip algorithm in Xiao and Boyd (2004) uses a communication mixing matrix $\mathbf{W} = [w_{i,j}] \in \mathbb{R}^{n \times n}$ to encode the topology of the network and communication weights along edges. At each step, gossip algorithm takes the local weighted average as the estimate of the global average and agents pass it to neighbours, i.e.,

$$x_i^{k+1} = \sum_{j \in \mathcal{N}_i} w_{i,j} x_j^k, \tag{2}$$

where $\mathcal{N}_i$ is the neighbour index set.

It has been shown that each x_i goes to the average linearly dependent on the connectivity of the network. The construction of $\mathbf{W}$ can be done using Laplacian matrix of G, which intuitively explains the dependence. From problem (1), $\mathbf{W}$ plays the important role in gossip algorithm and consensus is guaranteed. These two aspects direct further development of decentralized algorithms on more general problems.

Decentralized problem widely arises and is applied in many fields: decentralized state estimation of smart grid, decentralized dictionary learning, decentralized signal processing, etc. Compared with centralized system, it's more robust to the connectivity of network. The computation is also balanced because each node is treated equally. In some specific application scenarios, data privacy makes decentralization necessary.

The most general problem aims to find

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathbb{R}^{p \times n}}{\arg\min} \, \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} f_i(x_i) + g_i(x_i) \text{ subject to } \mathbf{W}\mathbf{x} = \mathbf{x} \tag{3}$$

where $\mathbf{x} = [x_1, \cdots, x_n]$, f_i and g_i are convex and privately known by agent i. f_i is smooth and g_i is not necessarily smooth but should be easily evaluated, e.g., l_1-regularizer. To make sure the consensus, i.e., $\mathbf{x}^* = x^*\mathbf{1}^T$, it's sufficient to construct $\mathbf{W}$ such that the largest eigenvalue of $\mathbf{W}$ is 1 and the corresponding eigenspace is spanned by $\mathbf{1}$.

Early decentralized algorithms based on (sub)gradient descent (e.g. Nedić and Ozdaglar, 2009) have sublinear convergence under strongly convex assumption when $g_i = 0$ and the diminishing step size is needed to get consensus, which creates the gap between decentralized algorithms and centralized ones. The ADMM (Alternate Direction Multiplier Method)-type decentralized method (e.g. Shi et al., 2014) is proven to have linear convergence in the same case. After that, many linearly convergent algorithms are proposed without g_i. Some examples are EXTRA (EXact firsT ordeR Algorithm) in Shi et al. (2015a), NIDS (Network InDependent Step-sizes Algorithm) in Li et al. (2017). When there is nonsmooth term g_i, PG-EXTRA in Shi et al. (2015b) and NIDS only have sublinear convergence rate, which is worse than the linear rate of method proposed in Alghunaim et al. (2019) although $g_i = g_j$ is assumed, i.e., a common regularizer will be shared.

Other interesting topics on decentralized optimization include directed decentralization (directed network), dynamic decentralization (dynamic network), communication compression (quantized information), optimizing graph related matrix to accelerate algorithm (optimal communication weights), etc. Some topics are quite challenging, e.g., asynchronous decentralized algorithm. My current and future research are based on these topics, which will be described in the following sections.

1 Theoretical Improvement

EXTRA is a well-known decentralized algorithm to solve problem (3) when there is no regularizer g_i. A relatively small upper bound of step-size depending on network and condition number of objective is required to attain linear convergence. Although a larger upper bound can be chosen in numerical experiments, no theoretical proof is given.

NIDS follows the very similar iterations to EXTRA except that each node needs to share local gradient information with its neighbours while in EXTRA, gradient will not be exchanged. This tiny difference makes NIDS can take network-independent step-size to achieve linear convergence under the same assumptions. In particular, NIDS can take the upper bound as large as that in centralized algorithms which closes the gap on step-sizes.

I have completed some theoretical results on both algorithms in Li and Yan (2019). The motivations are based on the following two aspects:

- Both of two algorithms rely on strongly convex assumption on $\mathbf{f}$, which is fairly strong and some basic problems don't satisfy, e.g., decentralized sensing problem, $f_i = \frac{1}{2}\|M_i x_i - b_i\|^2$, where $M_i \in \mathbb{R}^{1\times p}$, $p \geq 2$. A recent work Alghunaim et al. (2019) gives the proof of EXTRA taking larger step-size but still assumes the same strong assumptions. Hence it's worth considering to weaken the assumption on objectives to cover a wider range of problems.

- Another common assumption is on communication matrix $\mathbf{W}$ that the eigenvalues lie in $(-1, 1]$. A recent work Li and Yan (2017) gives a relaxed assumption on $\mathbf{W}$ that eigenvalues are in $(-\frac{4}{3}, 1]$, which derives the optimal upper bound of step-size for NIDS. Due to the similarity of two algorithms, relaxed $\mathbf{W}$ may be also doable on EXTRA and the largest upper bound may exist.

My finished work proves that both algorithms using relaxed $\mathbf{W}$ under the weakest assumption on objective still achieve linear convergence. In numerical experiments(Figure 1), the largest upper bound of step-size may give the convergence improvement on EXTRA. For NIDS, the network-independent step-size will be kept and the encouraging improvement can be observed if we relax $\mathbf{W}$.

This work closes the theoretical gap on two algorithms. Future research may focus on finding improved NIDS to have linear convergence with the existence of nonsmooth regularizer. NIDS on directed or dynamic network is worth considering but it's also challenging.

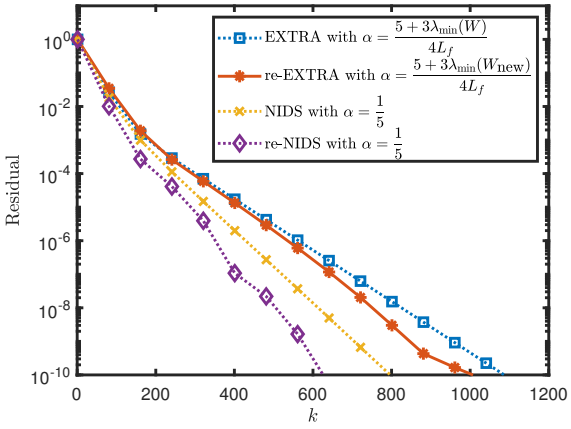

Figure 1: The improvement on EXTRA and NIDS.

2 Communication Compression

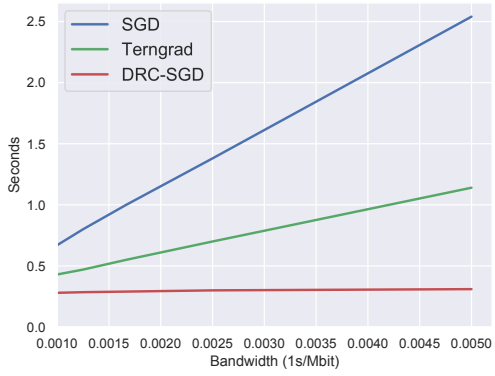

Figure 2: Time cost per iteration (Resnet18/CIFAR10).

In modern large-scale machine models, parallel stochastic gradient descent (PSGD) is used to do efficient training. The master aggregates gradient from workers and passes the updated model parameter back to each worker. It's, however, two-way communication that has become the bottleneck of efficiency as the number of worker and the dimension of model scale.

One common way to reduce the communication cost is to use compressed information under some quantization strategies. There are existing works (e.g. Alistarh et al., 2017; Wu et al., 2018; Tang et al., 2019) that propose quantized algorithm to compress data directly, error compensated data or the residual between the data and some reference parameter. Besides, some deterministic or stochastic quantization functions can be applied in each strategy. If we consider compressing data in both directions, more than 95% communication cost can be reduced depending on the specific quantization strategy and function we use. Even

sometimes the convergence will be slower than baselines, the final result on running time is still promising.

My recent work considers applying stochastic quantization function to residuals of gradient and model parameter with respect to some reference in both communication directions of PSGD. Figure 2 shows the superior result on Resnet18 over CIFAR10.

There are a few works now considering quantized decentralized algorithm. Some numerical experiments imply that the convergence of some algorithms is demanding on some specific quantization function. The training data is also required to be distributed evenly across nodes. Hence there is still an open area to do design better algorithms, which is beneficial to deep learning and decentralized optimization.

3 Acceleration

This work is about my ongoing project of ADMM. Vanilla ADMM can solve convex optimization problems with separable objective functions and linear constraints. It has $O(\frac{1}{k})$ sublinear convergence rate under general convex assumption on objectives. The result can be improved to be linear if one component of objective is strongly convex, which is proven in Giselsson and Boyd (2017). In this paper, authors propose the precondition trick on ADMM to obtain acceleration.We can compute precondition matrices of linear constraint to reduce the condition number which affects the final convergence and speed the algorithm up.

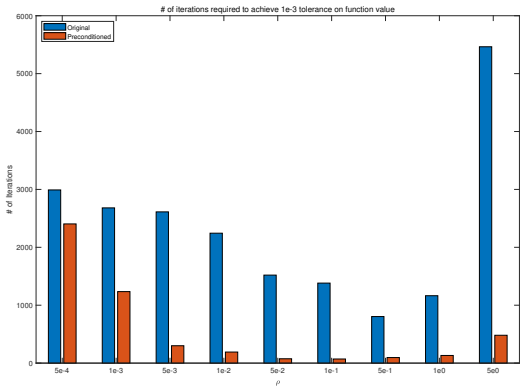

Figure 3: Iteration comparison between vanilla ADMM and preconditioned one.

This project aims to use the same trick in general convex case and design precondition method to get promising results on convolutional sparse coding problems. Figure 3 shows the promising result on basis pursuit denoising problem with mask decoupling.

My future research plan on this area is to to adapt this precondition trick on some decentralized algorithms. More specifically, since the communication matrix $\mathbf{W}$ is constructed by using Laplacian matrix, it may not be the optimal for the given network. If we can find precondition matrix of $\mathbf{W}$ which can help reduce the critical parameter affecting final convergence, the algorithm can be accelerated under the optimized communication weights.

References

Alghunaim, S. A., K. Yuan, and A. H. Sayed, 2019: A linearly convergent proximal gradient algorithm for decentralized optimization. *arXiv preprint arXiv:1905.07996*.

Alistarh, D., D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, 2017: Qsgd: Communication-efficient SGD via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pp. 1709–1720.

Giselsson, P. and S. Boyd, 2017: Linear convergence and metric selection for Douglas-Rachford splitting and ADMM. *IEEE Transactions on Automatic Control*, **62(2)**, 532–544.

Li, Y. and M. Yan, 2019: On linear convergence of two decentralized algorithms. *arXiv preprint arXiv:1906.07225*.

Li, Z., W. Shi, and M. Yan, 2017: A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *arXiv preprint arXiv:1704.07807*.

Li, Z. and M. Yan, 2017: A primal-dual algorithm with optimal stepsizes and its application in decentralized consensus optimization. *arXiv preprint arXiv:1711.06785*.

Nedić, A. and A. Ozdaglar, 2009: Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, **54**, 48–61.

Shi, W., Q. Ling, G. Wu, and W. Yin, 2015a: EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, **25(2)**, 944–966.

—, 2015b: A proximal gradient algorithm for decentralized composite optimization. *IEEE Transactions on Signal Processing*, **63(22)**, 6013–6023.

Shi, W., Q. Ling, K. Yuan, G. Wu, and W. Yin, 2014: On the linear convergence of the ADMM in decentralized consensus optimization. *IEEE Transactions on Signal Processing*, **62(7)**, 1750–1761.

Tang, H., X. Lian, T. Zhang, and J. Liu, 2019: Doublesqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression. *arXiv preprint arXiv:1905.05957*.

Tsitsiklis, J. N., 1984: Problems in decentralized decision making and computation. Tech. rep., Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems.

Wu, J., W. Huang, J. Huang, and T. Zhang, 2018: Error compensated quantized SGD and its applications to large-scale distributed optimization. In *ICML*.

Xiao, L. and S. Boyd, 2004: Fast linear iterations for distributed averaging. *Systems & Control Letters*, **53(1)**, 65–78.