MACHINE LEARNING OVER NETWORKS

Computer Assignment 5

Group 6

Milad Gangalizadeh
Hamid Ghourchian
Afsaneh Mahmoudi
Meysam Masoudi

1. Preliminaries

The cost function is

$$f(\mathbf{w}) = \frac{1}{m}\sum_{j=1}^{m} \tilde{f}_j(\mathbf{w}) + \lambda \frac{\|\mathbf{w}\|_2^2}{2} = \frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{w}),$$

where

$$f_i(\mathbf{w}) = \frac{N}{m} \sum_{j=(i-1)\frac{m}{N}+1}^{i\frac{m}{N}} \tilde{f}_j(\mathbf{w}) + \lambda \frac{\|\mathbf{w}\|_2^2}{2}, \;\; i = 1, \dots, N$$

$$\tilde{f}_j(\mathbf{w}) = \log\left(1 + e^{-y_j(\mathbf{x}_j^{\mathrm{T}}\mathbf{w})}\right), \;\; j = 1, \dots, m$$

$$\mathbf{x}_j \in \mathbb{R}^n, \;\; y_j \in \mathbb{R}.$$

As we proved in CA2, we have that $f_i(w)$ are L_i-Lipschitz and μ-strongly convex with

$$L_i \le \lambda + \frac{N}{4m}\lambda_{\max}\left(\sum_{j=(i-1)\frac{m}{N}+1}^{i\frac{m}{N}} y_j^2 \mathbf{x}_j \mathbf{x}_j^{\mathrm{T}}\right) \le L,$$

$$\mu \ge \lambda.$$

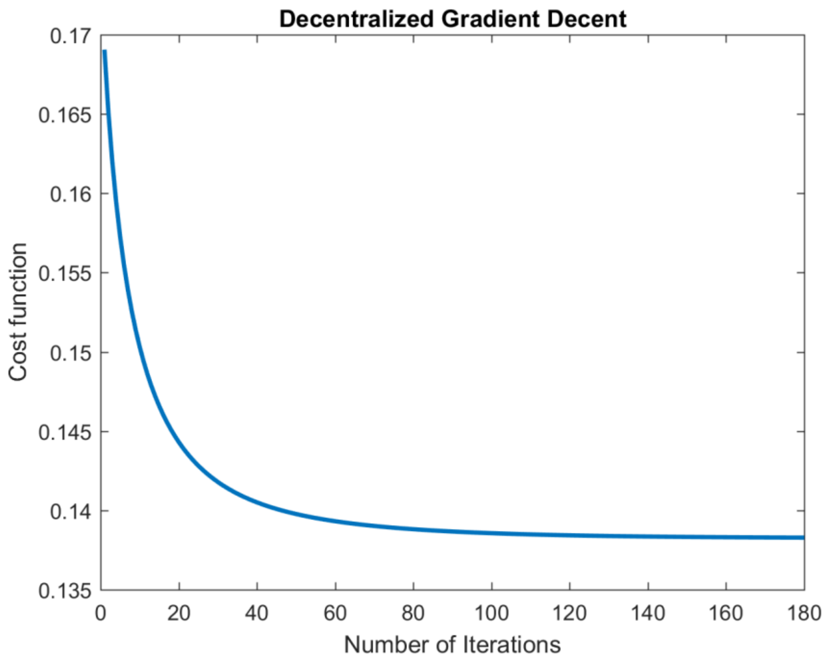
In the simulations, when the algorithms stop, $\|\nabla f(\mathbf{w})\|_2 \le \epsilon$. The dataset we used, $\{(\mathbf{x}_j, y_j)\}_{j=1}^{m}$, is MNIST, and the parameters are:

Number of features (n)	784
Number of samples (m)	60000
Number of partitions (N)	10
λ	1
ϵ	0.01
L	50
μ	1

We would like to find the optimal solution of the problem in a distributed manner, with N different nodes. Next, we explain different methods to do so.

2. Decentralized Gradient Decent (Master-Worker)

In this method, there are N workers and one master node. Hence, in each iteration, $\boxed{2N = 20}$ vectors of length n are communicated. In the simulation, we chose $\boxed{\alpha = 1/L = 0.01}$.



The time it takes is about $\boxed{57\text{ sec}}$, the number of iterations are $\boxed{T = 180}$, and the optimal solution is approximately $\boxed{0.1383}$.

One way to improve the number of communications in each iteration is to use stochastic gradient decent instead of gradient decent. Although in theory, the rate of convergence of SGD is sublinear; so T may become greater than GD.

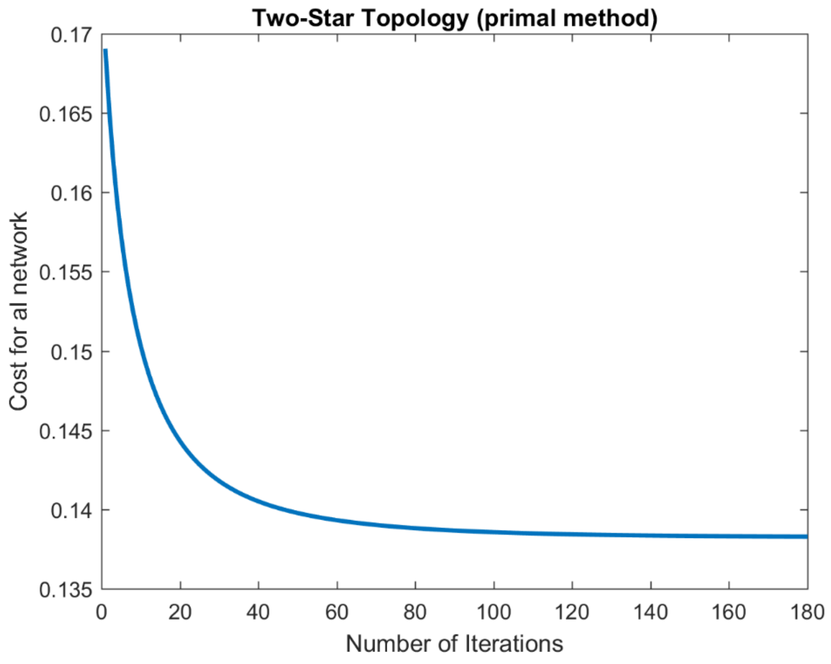
3. Two-star Topology

a. Decentralized sub-gradient (primal method, v1)

In this method, we have used the following doubly stochastic matrix:

$$A = \begin{bmatrix} 0.8 & 0 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.8 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.8 & 0 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0 & 0 & 0 & 0 & 0 & 0.2 & 0.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.2 & 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.2 & 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0.8 \end{bmatrix}$$

The second greatest eigenvalue of A is $\sigma_2 = 0.94$. The number of communications in each iteration is twice as the number of edges of the graph, which is $\boxed{18}$, and in each one, a vector of length n is communicated.



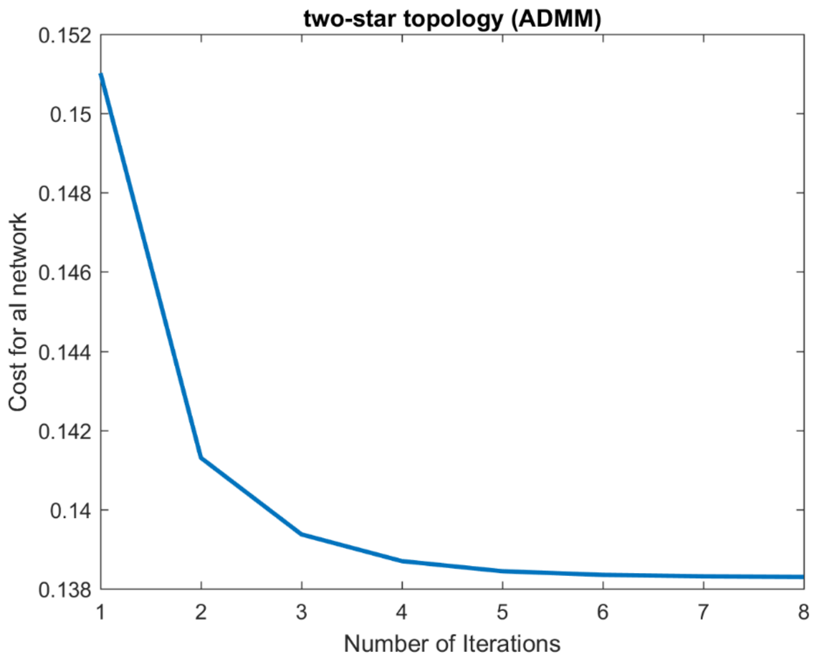
The time it takes is about $\boxed{60 \text{ sec}}$, the number of iterations are $\boxed{T = 180}$, and the optimal solution is approximately $\boxed{0.1383}$. Hence, this method has 360 communication less than master-worker method. The value of the objective function is evaluated based on the average of the variables of all nodes.

Again, we can use stochastic GD instead of GD in each node. Even for the consensus part, we can randomly select one of the values of the neighbourhood nodes according to the probability distribution of the corresponding row of matrix A.

b. ADMM over Networks

We chose $\rho = 1$ in the problem. Similar to the previous method, the number of communications in each iteration is twice as the number of edges of the graph, which is $\boxed{2N = 18}$. In each node, we have used gradient decent method with step size $\alpha = 0.01$ to find the optimal primal variable.

The time it takes is about $\boxed{29\ sec}$, the number of iterations are $\boxed{T = 8}$ (apart from the number of iterations in each node), and the optimal solution is approximately $\boxed{0.1383}$. The value of the objective function is evaluated based on the average of the variables of all nodes.



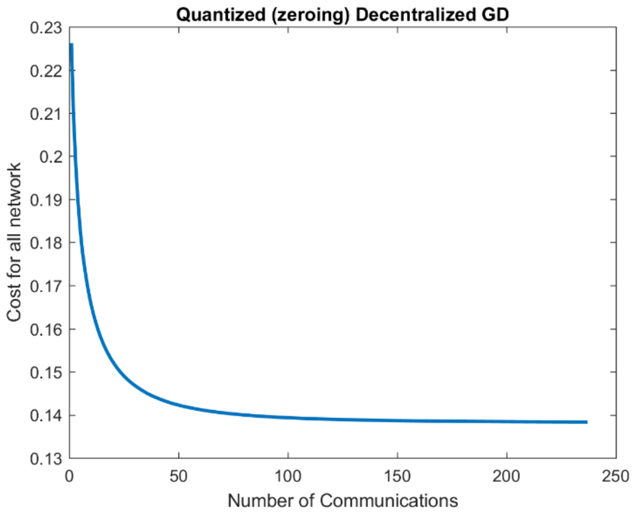
In order to improve the number of communications in this method, one can only use the value of one its neighbours with uniform probability. One drawback of this method is that the rate of the convergence may become sublinear.

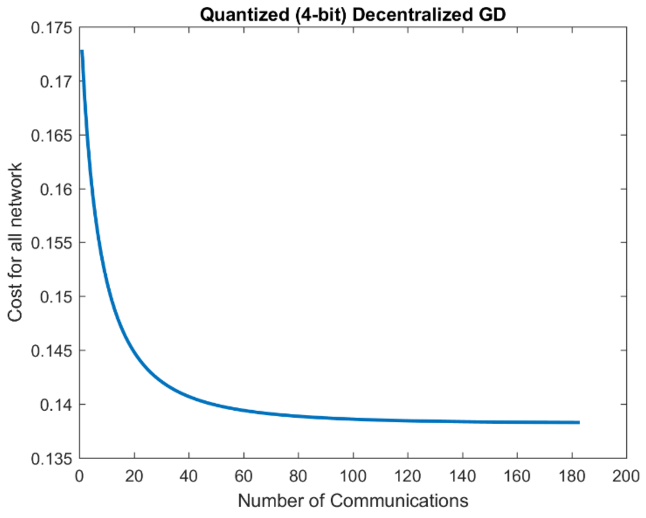
4. Quantization

In the simulations, for zeroing method, the probability of an element of a vector being 0 is 0.6, and the number of bits sent is 4.

a. Master-worker

	Zeroing	**4-bit**
Number of iterations	237	183
Final value	0.1383	0.1383
Time (sec)	77	60





b. Two-Star Topology

i) Subgradient (primal)

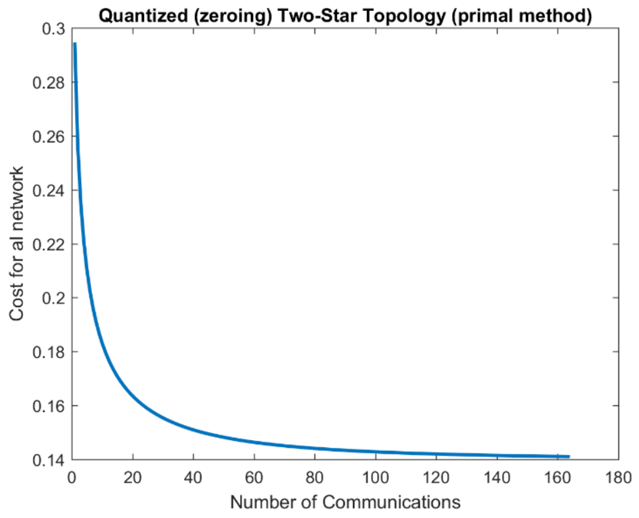
We used Quantized Distributed Gradient Descent (QDGD) algorithm with $\delta = 0.4$, maximum number of iterations, $M = 1000$, as well as,

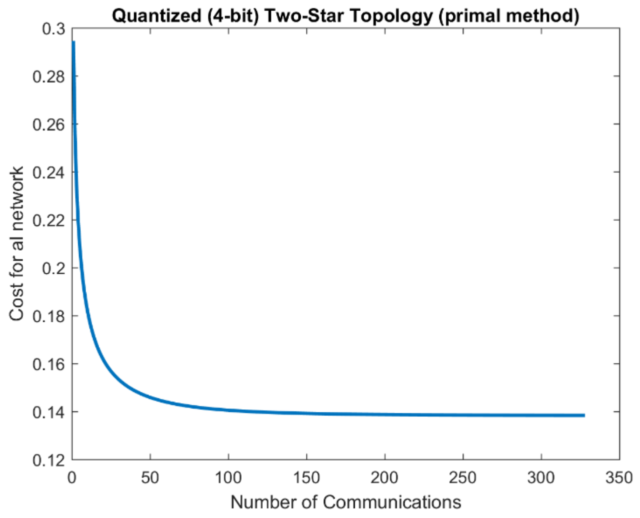
$$\epsilon = \frac{1}{M^{1.5\delta}}, \quad \alpha = \frac{1}{M^{0.5}}.$$

	Zeroing	**4-bit**
Number of iterations	164	328
Final value	0.1411	0.1384
Time (sec)	51	106

ii) ADMM

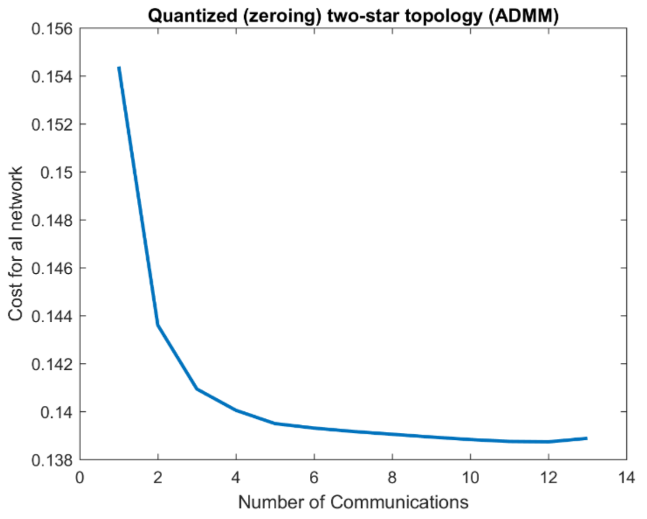
For quantized distributed ADMM, we used almost the algorithm of (Zhu, et al., 2015), which each node uses the quantized value of its neighbourhood. Regarding

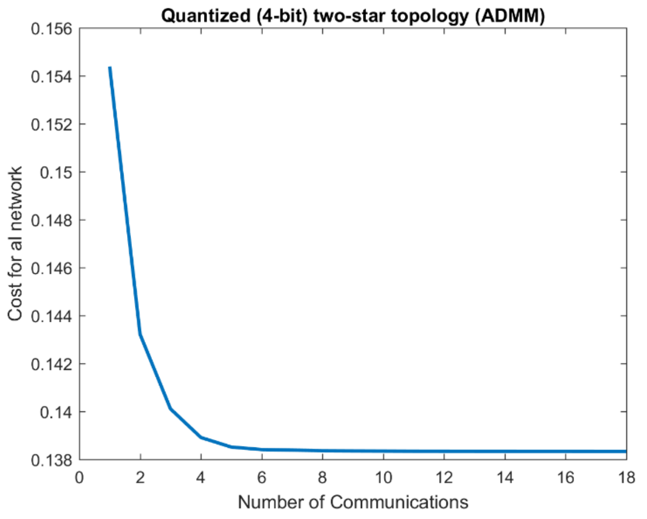




the value of itself, it uses the quantized version of its own value for updating the dual variable, and the original value for updating the primal variable.

	Zeroing	4-bit
Number of iterations	13	18
Final value	0.1389	0.1383
Time (sec)	56	85





c. Combining Quantization Methods

Two methods can be combined into one, i.e., we only consider k elements in each communication, and sending only 4 bits of each of k elements. Note that each of two methods are equivalent to SGD and the convergence can be proved using the theorems of SGD.

5. SVRG and SAG

In order to use these methods in a distributed way, one way is to use federated learning which uses SVRG in each node, and then send the value of the variable to its neighbours. In ADMM, the internal optimization can be solved by SVRG or SAG.

6. References

Zhu, Shengyu, Hong, Mingyi and Chen, Biao. 2015. Quantized Consensus ADMM for Multi-Agent Distributed Optimization. 2015.