Probabilistic DIGing

I. Brief Description

The optimization problem is given by

$$\textbf{(P1)} \quad \mathbf{\Theta}^* := \arg\min_{\mathbf{\Theta}} \sum_{n=1}^{N} f_n(\mathbf{\Theta}), \tag{1}$$

where $\mathbf{\Theta} \in \mathbb{R}^{d\times 1}$ is the model parameter and $f_n : \mathbb{R}^d \to \mathbb{R}$ is a local function composed of data stored at worker n.

A. Linear Regression

1) Loss Function: In this case, the local cost function at worker n is explicitly given by

$$f_n(\boldsymbol{\theta}) = \frac{1}{2}\|\boldsymbol{X}_n\boldsymbol{\theta} - \boldsymbol{y}_n\|^2, \tag{2}$$

where $\boldsymbol{X}_n \in \mathbb{R}^{s\times d}$ and $\boldsymbol{y}_n \in \mathbb{R}^{s\times 1}$ are private for each worker $n \in \mathcal{V}$ where s represents the size of the data at each worker.

2) Datasets: In this task, we will consider the following datasets (see Table 2 of http://cacr.uwaterloo.ca/techreports/2019/cacr2019-05.pdf):

- Boston: https://github.com/benchopt/benchmark_ols/tree/master/datasets
- Wine Quality: https://archive.ics.uci.edu/ml/datasets/wine+quality

B. Logistic Regression

1) Loss Function: In this subsection, we consider the L_2-regularized binary logistic regression task. We assume that each worker n owns a data matrix $\boldsymbol{X}_n = (\boldsymbol{x}_{n,1}, \ldots, \boldsymbol{x}_{n,s})^T \in \mathbb{R}^{s\times d}$ along with the corresponding labels vector $\boldsymbol{y}_n = (y_{n,1}, \ldots, y_{n,s}) \in \{-1, 1\}^s$. The local cost function for worker n is then given by

$$f_n(\boldsymbol{\theta}) = \frac{1}{s}\sum_{j=1}^{s} \log\left(1 + \exp\left(-y_{n,j}\boldsymbol{x}_{n,j}^T\boldsymbol{\theta}\right)\right) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2, \tag{3}$$

where λ is the regularization parameter.

2) Datasets: In this task, we will consider the following datasets:

- a1a: https://github.com/konstmish/opt_methods/tree/master/datasets
- mushrooms: https://github.com/konstmish/opt_methods/tree/master/datasets
- w8a: https://github.com/konstmish/opt_methods/tree/master/datasets
- madelon: https://github.com/benchopt/benchmark_logreg_l2/tree/master/datasets
- covtype: https://github.com/benchopt/benchmark_logreg_l2/tree/master/datasets

Dataset	**Task**	**Model Size (d)**	**Number of Instances**	**Number of Workers (N)**
	linear regression			
	linear regression			
	logistic regression			
	logistic regression			

TABLE I: List of datasets used in the numerical experiments.

C. Websites for datasets

- https://www.openml.org/
- https://archive.ics.uci.edu/ml/datasets.php

D. Algorithms

1) Decentralized SGD:

- Initialization: $\boldsymbol{\theta}^0 \in \mathbb{R}^d$.
- Model Update

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{W}^k \boldsymbol{\theta}^k - \alpha \nabla \boldsymbol{f}(\boldsymbol{\theta}^k) \tag{4}$$

2) DIGing:

- Initialization: $\boldsymbol{\theta}^0 \in \mathbb{R}^d$, $\boldsymbol{\delta}_n^0 = \nabla f_n(\boldsymbol{\theta}^0)$.
- Model Update

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{W}^k \boldsymbol{\theta}^k - \alpha \boldsymbol{\delta}^k \tag{5}$$

$$\boldsymbol{\delta}^{k+1} = \boldsymbol{W}^k \boldsymbol{\delta}^k + \nabla \boldsymbol{f}(\boldsymbol{\theta}^{k+1}) - \nabla \boldsymbol{f}(\boldsymbol{\theta}^k) \tag{6}$$

3) Proposed: Probabilistic DIGing:

- Initialization: $\boldsymbol{\theta}^0 \in \mathbb{R}^d$, $\boldsymbol{\delta}_n^0 = \nabla f_n(\boldsymbol{\theta}^0)$.
- Model Update

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{W}^k \boldsymbol{\theta}^k - \alpha \boldsymbol{\delta}^k \tag{7}$$

- Gradient Tracking

$$\boldsymbol{\delta}^{k+1} = \begin{cases} \nabla \boldsymbol{f}(\boldsymbol{\theta}^{k+1}), & \text{with probability } p^k \\ \boldsymbol{W}^k \boldsymbol{\delta}^k + \nabla \boldsymbol{f}(\boldsymbol{\theta}^{k+1}) - \nabla \boldsymbol{f}(\boldsymbol{\theta}^k), & \text{with probability } 1 - p^k \end{cases} \tag{8}$$

E. Tasks

- Dataset preparation: download at least 4 datasets for each task (regression/classification) in the format X and y (as .npy files if possible, otherwise any format you are comfortable with).
- Strongly convex case
  - Start simple: implement DGD and DIGing for the linear regression + logistic regression cases.
  - Implement the probabilistic DIGing and experiment with different choice of the probability p.
    (i) $p^k = \frac{a}{a+k}, \quad a > 0.$
    (ii) $p^k = \exp\left(-\frac{k}{T}\right), \quad T > 0.$
  - Plots (need to think more about it):
    (i) train/test loss (accuracy) or residuals vs. number of iterations.
    (ii) train/test loss (accuracy) or residuals vs. cumulative communication cost. How to define the cumulative communication cost in our context?

$$C_T^k = C_T^{k-1} + \sum_{n=1}^{N} d_n c^k \tag{9}$$

where c^k is the size of the variables exchanged at iteration k and the residuals (R) are defined as

$$R^k = \frac{\|\boldsymbol{\theta}^k - \boldsymbol{\theta}^\star\|_F}{\|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^\star\|_F} \tag{10}$$

- Stochastic version of the algorithms.
- Proof attempt (strongly convex/dynamic, non-convex static/ non-convex dynamic).