CS-439 Project - Distributed Learning

Adrian Sager La Ganga (327960), Eelis Mielonen (323681), Maximo Cravero (321435)
Department of Computational Science & Engineering, EPFL, Switzerland

I. Introduction

In the big-data era, distributed learning is becoming more and more important as the increasing size of data-sets prohibits them from being processed on a single machine. Data privacy concerns also place limitations on learning schemes where data is shared between participants. As a result, distributed learning techniques where model weights are shared between participants are currently a hot research topic.

The benefit of distributing model training is that we can partition large data-sets or models across nodes, and exploit parallel computation in order to find the optimum faster. However, this comes with a host of new problems. For instance, if we partition data-sets across nodes in a biased way, models that are trained on these nodes will reflect that bias, and more computational effort is required to correct for it. This issue could be solved by simply sending the model weights to all other participating nodes and computing the average, however this is not practical given that distributed systems have limited bandwidth.

Therefore we want nodes to communicate with each other as little as possible, but we also want to obtain a globally accurate model. There are multiple facets of this problem that can be explored, but in this report, we will focus on the effect that different network topologies have on the convergence rate of GD on a strongly convex and smooth objective. We will simulate distributed learning on a synthetic, biased data-set comparing the convergence rate in relation to theory, and measure the total amount of data exchanged using different topologies.

II. Background

A. Distributed Averaging

In distributed learning, we partition the dataset D into equally sized subsets D_i between nodes indexed by i. Each node then instantiates their own model weights w_i^0. After this, training is performed in two alternating steps for a certain amount of total iterations. First, each node computes the gradient with respect to D_i and their weights w_i^t, and performs a local GD or SGD update to obtain $w_i^{t+1/2}$. Secondly, these updated weights are mixed between neighboring nodes to obtain the weights for the next iteration w_i^{t+1}. More specifically [1]:

$$w_i^{t+1} = \sum_{j=1}^{N_{nodes}} W_{ij} w_j^{t+1/2}$$

Where W is a symmetric, non-negative, doubly stochastic matrix, and $W_{ij} \neq 0$ iff. i and j are neighboring nodes. Computing an exact average of the weights over all nodes, with a fully connected topology, corresponds to choosing $W_{ij} = \frac{1}{N_{nodes}}$. The main complication comes in when we want to limit the number of communications, thus the number of nonzero entries in W. With a sparse topology we can obtain an approximation of the global average with repeated mixing steps, since we have that $\lim_{t\to\infty} W^t \to W_{fc}$, where W_{fc} is the fully connected weight matrix. In other words, with nodes sharing weights with each other multiple times, we can approximate the global average. One particular way of building a valid W is with the metropolis hastings (MH) weight assignment:

$$W_{ij} = \begin{cases} \frac{1}{1+\max(\deg(i),\deg(i))}, & i \neq j \\ 1 - \sum_{i\neq j}^{N_{nodes}} W_{ij}, & i = j \end{cases}$$

Where $\deg(i)$ refers to the degree of the node i in the graph of connections. Other more efficient weight assignments exist [2], but in this project we chose to use this method because of its simplicity.

III. Models & Methods

A. Dataset

As previously stated, we constructed a synthetic dataset to explore the effect of non-IID data between nodes. In particular, our dataset is composed of $N := 2000$ 2-dimensional points $X \in \mathbb{R}^{N\times 2}$, and their targets $y \in \mathbb{R}^{N\times 1}$. The points in X were sampled as follows:

$$X_{i,1} \sim \mathcal{N}(0,1) \quad X_{i,2} \sim \text{abs}\left(\mathcal{N}(0,1)\right) \quad i = 1,\ldots,N$$

Thus, all points lie on a semi-Gaussian. The targets were calculated from X:

$$y_i := \exp\left(-\|X_{i,:}\|_2^2\right) - \frac{1}{4} \qquad i = 1,\ldots,N$$

Adding $-1/4$ centers the targets. These points were partitioned into equally sized pieces for each node, as in Figure 1. Because the targets in the center of the distribution have higher values, fitting a plane gives biased results at each node.

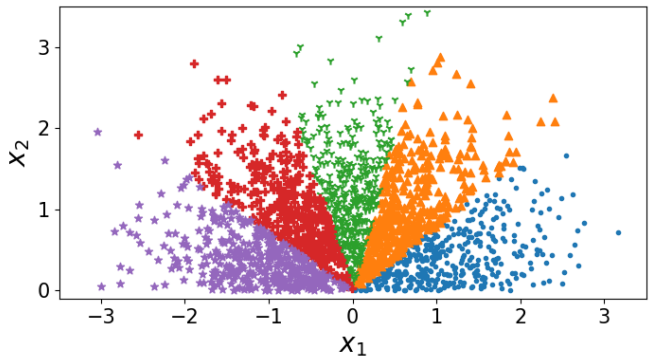

Fig. 1. Data distributed to 5 nodes, each corresponding to a color.

B. Model and Loss Function

We optimized a linear regression model with the Mean Squared Error (MSE) loss function:

$$f(w,b) := \frac{1}{N}\left\|y - (X \cdot w + b)\right\|_2^2 \tag{1}$$

Where $w := (w_1, w_2) \in \mathbb{R}^{2\times 1}$ are the weights and $b \in \mathbb{R}$ is the bias. For simplicity, we defined $w' := (w_1,\ w_2,\ b)$, so $f(w') = f(w,b)$. f in our case is μ-strongly convex and L-smooth. Concretely,

$$\mu = \frac{2}{N}\lambda_{\min}\left((X')^T \cdot X'\right) \leq L = \frac{2}{N}\lambda_{\max}\left((X')^T \cdot X'\right)$$

Where X' is X with an additional column of ones to account for biases, and $\lambda_{\min}$, $\lambda_{\max}$ are the minimum and maximum eigenvalues. For our dataset, $\mu = 0.3879$ and $L = 3.5575$. As a result of these properties, we have a unique global minimum w'_*. Additionally, the value of f at w'_* can be easily computed with the normal equations, and in our case it is $f(w'_*) = 0.05307$. The loss $f_i(w'_i)$ computed at node i with its parameters w'_i is equivalent to Equation 1. In our results section, we report the mean MSE over all nodes obtained by applying each w'_i to the global dataset (X', y).

C. Topologies

We considered 4 different topologies in this project.

1) **Fully-connected (FC)**: $W_{i,j} = 1/N_{\text{nodes}} \quad \forall i,j$.
2) **Ring**: Nodes are connected into a ring, where each node has two neighbors. $W_{i,j} = 1/3$, $j \in \{i-1, i, i+1\} \mod N_{\text{nodes}}$.
3) **Random**: The weights in W_{ij} are set to 1 with probability p, ensuring symmetry by computing $W+W^T$, and then setting the nonzero weights correctly with the MH assignment.
4) **Small World (Watts-Strogatz)**: Form a ring, connect each node to k nearest neighbors, and replace every edge with a random edge with probability p. Then set the nonzero weights with MH.

According to [3], the convergence of the loss function $f(w,b)$ in a distributed setting can be determined under certain simple assumptions, namely the convexity of the objective, and synchronicity in the distributed updates. It is shown that given a global learning rate $\gamma \leq \min\{\frac{1+\lambda_n(W)}{L_h}, \frac{1}{\mu_{\bar{f}}+L_{\bar{f}}}\}$, where $L_h = \max_{i\in n}\{L_{f_i}\}$, $L_{\bar{f}} = \frac{1}{n}\sum_{i=1}^n L_{f_i}$, $\mu_{\bar{f}} = \frac{1}{n}\sum_{i=1}^n \mu_{f_i}$, and f_i is the loss at node i, we can guarantee a linear rate of convergence. Furthermore, the authors provide a result which relates the spectral properties of W with how close to the global optimum each model in the network can converge to. It is stated that, given a particular mixing matrix W, all local solutions converge to within $O(\frac{\gamma}{1-\beta})$ of the global optimum, where $\beta = \max\{|\lambda_2(W)|, |\lambda_n(W)|\}$, $\lambda_2(W)$ is the second largest eigenvalue, $\lambda_n(W)$ the smallest. For this project, we computed β for each topology and selected γ based on the rule outlined above. This provided a fair comparison between topologies, and allowed us to study the correspondence between the empirical convergence and theory, but now with non-IID data.

IV. Results

As stated above, for gradient descent with a smooth and strongly convex function, we expect to obtain linear convergence given the right γ. We observed that for a FC topology with any number of nodes, and 1 mixing step between iterations, we get the expected rate. Under the same conditions, all the random topologies converged linearly, until they plateau to a certain error threshold. This coincides well with the theoretical results.

For instance, in Fig. 2 we see that lower β corresponds to a lower final loss reached, seemingly within $O\left(\frac{\gamma}{1-\beta}\right)$ of the optimum. Here, we fixed γ to isolate the effect of β.

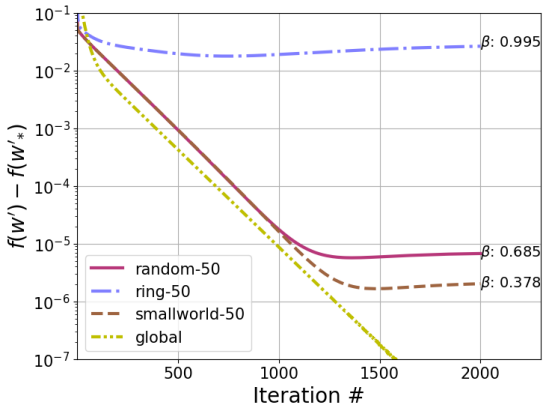

Fig. 2. Convergence plots for different topologies. $\gamma = 0.01$. *Global* is the convergence plot for non-distributed GD on the objective.

When comparing topologies in terms of the required node-to-node communications to reach a certain tolerance, we observe that with simple topologies and MH weight assignment we can only make minor savings in communications to reach a certain error, compared to the FC

case. For instance, to reach an error of the order 10^{-3} a random topology requires around 50% of the information exchanged in FC, and to reach an error of the order 10^{-4} we can save about 30% of information using the small world topology. This can be seen in Fig. 3, where communications are measured by total number of bytes shared between the nodes. γ is set to its theoretical upper bound for each topology to provide a fair comparison, since higher γ causes faster convergence, and thus less total bytes for the same error tolerance.

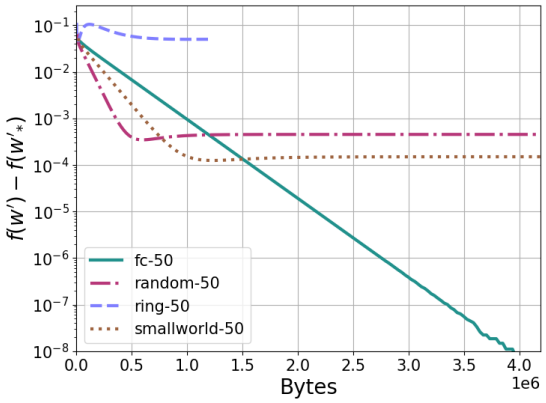

Fig. 3. Error vs. total bytes exchanged for different topologies. 50 nodes, 1 mixing step per iteration. k = 25 for the Small World topology.

Additionally, we observed that breaking the 10^{-4} error threshold through more mixing steps requires more total communications than the FC topology, as seen in Fig. 4. In the same Figure, we can see a trade-off: more mixing steps imply lower error, but require more data exchange.

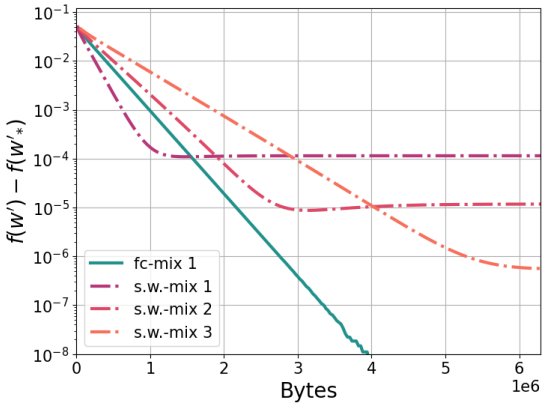

Fig. 4. Error vs. total bytes exchanged for different number of mixing steps for the Small World topology. $k = 25$. 50 nodes.

An important note is that in practice there are diminishing returns to achieving a higher accuracy level (like overfitting), and that it is reasonable to assume that the performance of a small world topology would be sufficient.

V. Discussion

The main point of discussion is the trade-off between the final achieved accuracy and the total cost of communication. As indicated in the plots, the convergence rate is linear up until the point that it reaches a neighborhood of the optimum, which depends on the β parameter.

We observed that for a biased dataset on the distributed linear regression, there is an inherent trade-off between sparsity, communications and the final accuracy reached. We saw that even though the data is non-IID, we can empirically obtain the same convergence rate as for the IID case.

A sparse topology may be needed in case nodes can't communicate with each other directly, or if communicating with some nodes requires a low bandwidth. This may be useful in real applications as a fully connected distributed server requires more maintenance and incurs higher costs. One caveat is that in practice we do not deal with synchronous systems, and the variance among models in different nodes introduced by inexact averaging could be compounded by this asynchronicity.

It should also be be noted that the Metropolis-Hastings weight assignment may not give the fastest mixing matrix [2]. A future study could investigate weight assignment methods that ensure symmetric and doubly stochastic mixing matrices that provide better spectral properties such that both higher learning rates and closer convergence to the optimum can be ensured.

Furthermore, in future work these optimal mixing weights could be tried on a higher dimensional dataset, like MNIST, with a more complex model, like a Neural Network, for a closer real-world example. We can also repeat the experiments with IID data, since the random topologies may actually perform better.

Based on our results, we see that we can leverage different properties of different topologies depending on accuracy requirements and system requirements, such as the communication budget. Another interesting area to explore would be hybrid topologies, where we can initially use a more sparse topology to obtain a certain level of accuracy at a very low cost, then adjust the neighborhood of each node to obtain a more dense topology and achieve higher accuracies.

References

[1] A. Nedić, A. Olshevsky, A. Ozdaglar, and J. Tsitsiklis, "On distributed averaging algorithms and quantization effects," 2009.

[2] S. Boyd, P. Diaconis, P. Parrilo, and L. Xiao, "Fastest mixing markov chain on graphs with symmetries," *SIAM Journal on Optimization*, vol. 20, no. 2, p. 792–819, Jan 2009. [Online]. Available: http://dx.doi.org/10.1137/070689413

[3] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," 2015.