Communication Avoiding Optimization Algorithm for Solving L1-regularized Problems

Saeed Soori, Ajinkya Padwad, Rachit Shah

November 8, 2017

1 Project Description

The iterative soft thresholding algorithm (ISTA) is used to solve convex regularized optimization problems in machine learning. Distributed implementations of the algorithm have become popular since they enable the analysis of large datasets. However, existing formulations of the algorithm communicate data at every iteration and thus have high communication overheads. The communication costs are closely tied to the mathematical formulation of the algorithm. In this work, we reformulate ISTA to communicate data in every "k" iterations and thus reduce the cost of communication in the algorithm when operating on large data sets. We formulate the algorithm for three different optimization methods on the Lasso problem and show that the latency cost is reduced by a factor of k while the bandwidth cost remains the same. The algorithm is re-organized so that to reduce synchronization steps while maintaining similar convergence rates and stability properties. The performance and scalability of the novel formulations for FISTA (Fast Iterative Thresholding Algorithm), and proximal newton type methods are demonstrated on a distributed hardware platform which attains speedups of up to 10x on Comet supercomputers.

2 Algorithm

Table 1 shows the CA-FISTA algorithm for solving Lasso problem. We reduce the communication for solving a regularized least square problem by a factor of $O(s)$ without changing the overall bandwidth and flop cost for K iteration. The communication avoiding algorithm communicates more words every s iteration, but it preserve the total bandwidth cost while keeping the convergence behavior intact. We unroll the update recurrences for fast iterative thresholding and stochastic proximal newton method by a factor of s, then compute the stochastic Hessian and residual for s iterations beforehand and send it to all processors. Then Every processor updates optimization variable using a soft thresholding operator.

Table 1: CA-FISTA Algorithm

Algorithm 3 Communication-Avoiding FISTA (CA-FISTA) Algorithm	
1:	**Input:** $X \in \mathbb{R}^{d \times n}$, $y \in \mathbb{R}^n$, $w_0 \in \mathbb{R}^d$, K>1, $b \in \mathbb{Z}_+$ s.t $b \leq n$
2:	**for** $k = 0, 1, ..., \frac{K}{s}$ do
3:	**for** $j = 1, ..., s$ do
4:	Generate a random matrix, $I_{sk+j} \in \mathbb{R}^{n \times b}$ with one non-zeros per column indicating samples used for computing gradient and Hessian.
5:	$G_j = \frac{1}{b} X I_{sk+j} I_{sk+j}^T X^T, R_j = \frac{1}{b} X I_{sk+j} I_{sk+j}^T y$
6:	set $G = [G_1 \vert G_2 \vert ... \vert G_s]$ and $R = [R_1 \vert R_2 \vert ... \vert R_s]$ and send them to all processors
7:	**for** $j = 1, ..., s$ do
8:	H_{sk+j} are $d \times d$ block of G
9:	$\nabla g(w_{sk+j}) = H_{sk+j} w_{sk+j-1} - R_{sk+j}$
10:	$w_{sk+j} = \underset{y}{\operatorname{argmin}} \nabla g(w_{sk+j})^T (y - w_{sk+j-1}) + \frac{1}{2}(y - w_{sk+j-1})^T (y - w_{sk+j-1}) + h(y)$ solve the optimization using FISTA:
11:	$v_{sk+j} = w_{sk+j-1} + \frac{sk+j-2}{sk+j}(\Delta w_{sk+j-1})$
12:	$w_{sk+j} = S_{\lambda t_k}(v_{sk+j} - t_k \nabla g(w_{sk+j}))$
13:	**output** w_K