Description and Implementation of a Distributed Gradient Method for Distribution Optimization With a Partial Variable With Induced Subgraphs

This document describes the direct application of the gradient method to the problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ \ f_1(x_{S_1}) + f_2(x_{S_2}) + \cdots + f_P(x_{S_P})\,, \tag{1}$$

where function f_p depends only on a subset $S_p \subseteq \{1,\ldots,n\}$ of components of the variable $x \in \mathbb{R}^n$. We will see, however, that this is efficient only in problems where each component x_l induces a star subgraph, for $l = 1,\ldots,n$. In other words, there is a connected network with P nodes, where the pth node knows only f_p; in this network, the set of nodes that depends on x_l is a star.

Derivation. Let $\mathcal{G} = (\mathcal{V},\mathcal{E})$ be the communication network where we want to solve (1) in a distributed way. Let $\mathcal{V}_l$ be the subgraph induced by x_l, i.e., the set of nodes in $\mathcal{V}$ whose function f_p depends on x_l ($l \in S_p$). We assume $\mathcal{V}_l$ is a star.

Denote the objective of (1) by $f(x)$. Its gradient w.r.t. x_l is

$$\frac{\partial}{\partial x_l} f(x) = \sum_{p\in\mathcal{V}_l} \frac{\partial}{\partial x_l} f_p(x_{S_p})\,.$$

The gradient method assumes $f(x)$ is convex, continuously differentiable, and its gradient $\nabla f(x)$ is Lipschitz continuous with constant L, i.e. $\|\nabla f(y) - \nabla f(x)\| \le L\|y - x\|$, for any x, y in the domain of f. Each component x_l is updated as

$$x_l^{k+1} = \Big[x_l^k - \frac{1}{L}\sum_{p\in\mathcal{V}_l} \frac{\partial}{\partial x_l} f_p(x_{S_p})\Big]_l^+\,, \tag{2}$$

where $[\cdot]_l^+$ is a projection operator for the component x_l. This is imposed by the node that is the center of the star $\mathcal{V}_l$, as explained next. In a distributed scenario, the update (2) has to take place at a given node. That node has to collect all the partial derivatives $\frac{\partial}{\partial x_l} f_p(x_{S_p})$, sum them, perform the update, and compute the projection. This is only efficient if the topology of $\mathcal{V}_l$ is a star and these operations are all performed at the center of the star. This is illustrated in Algorithm 1.

Algorithm 1 Distributed Gradient Method

Initialization: Each node that is a center for x_l, sets x_l^1 to an arbitrary value and sends in to the neighbors that depend on x_l; set $k = 1$

repeat
  for all $p = 1,\ldots,P$ [in parallel] **do**
    Compute $g_l^k := \frac{\partial}{\partial x_l} f_l(x_l^k)$ for all $l \in S_p$
    Send g_l^k to all neighbors that depend on x_l, i.e., $\mathcal{N}_p \cap \mathcal{V}_l$
  end for
  for all p such that p is the center of the star $\mathcal{V}_l$ [in parallel] **do**
    Update x_l as

$$x_l^{k+1} = \Big[x_l^k - \frac{1}{L}\sum_{p\in\mathcal{V}_l} \frac{\partial}{\partial x_l} f_p(x_{S_p})\Big]_l^+$$

    Send x_l^{k+1} to all neighbors that depend on x_l, i.e., $\mathcal{N}_p \cap \mathcal{V}_l$
  end for
  $k \leftarrow k + 1$
until some stopping criterion is met

In each k-iteration of Algorithm 1, there is information flowing in each edge, both ways just once. Thus, each iteration of the algorithm requires on communication step.