Description and Implementation of a 2-Block ADMM Algorithm for Problems with Star-Shaped Variables

We describe the algorithm in section 7.2 of

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," Found. Trends Mach. Learning, Vol.3, No.4, 2010

This algorithm solves

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

where function f_p depends only on a subset $S_p \subseteq \{1,\ldots,n\}$ of components of the variable $x \in \mathbb{R}^n$. However, as is stated in section 10.1 of [1], the algorithm proposed there requires a global aggregation mechanism, i.e., a central node where each node can, in one operation, broadcast a message to all the other nodes in the network. This is not distributed in our sense. For us, distributed means that, besides no central node, each node can only communicate with its neighbors. In this document we describe how the algorithm proposed in section 7.2 of [1] can be used to solve (1) in a distributed scenario. However, its implementation in a distributed scenario is only efficient, i.e., without requiring consensus steps within subgraphs, in a special case of the variable x in (1): each component x_l induces a subgraph that is a star, 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. Now, create a copy of x_l in all nodes in $\mathcal{V}_l$ and denote the copy at the pth node with $x_l^{(p)}$. We rewrite (1) as

$$\begin{array}{ll} \underset{\bar{x},z}{\text{minimize}} & f_1(x_{S_1}^{(1)}) + f_2(x_{S_2}^{(2)}) + \cdots + f_P(x_{S_P}^{(P)}) \\ \text{subject to} & x_l^{(p)} = z_l\,, \quad p \in \mathcal{V}_l, \quad l = 1,\ldots,n\,, \end{array} \tag{2}$$

where $x_{S_p}^{(p)} := \{x_l^{(p)}\}_{l\in S_p}$ is the set of all copies at node p. The variable is $(\bar{x}, z)$, where $\bar{x} = (\bar{x}_1,\ldots,\bar{x}_n)$, with $\bar{x}_l := \{x_l^{(p)}\}_{p\in\mathcal{V}_l}$ denoting all the copies of the component x_l, and $z \in \mathbb{R}^n$.

Next, we form the augmented Lagrangian of (2)

$$L_\rho(\bar{x}, z; \lambda) = \sum_{p=1}^{P} f_p(x_{S_p}^{(p)}) + \sum_{l=1}^{n}\sum_{p\in\mathcal{V}_l}\Big({\lambda_l^{(p)}}^\top (x_l^{(p)} - z_l) + \frac{\rho}{2}\|x_l^{(p)} - z_l\|^2\Big) \tag{3}$$

$$= \sum_{p=1}^{P} f_p(x_{S_p}^{(p)}) + \sum_{p=1}^{P}\sum_{l\in S_p}\Big({\lambda_l^{(p)}}^\top (x_l^{(p)} - z_l) + \frac{\rho}{2}\|x_l^{(p)} - z_l\|^2\Big)\,, \tag{4}$$

where $\lambda_l^{(p)}$ is the dual variable associated to the constraint $x_l^{(p)} = z_l$ in (1). According to the 2-block ADMM, we minimize (4) first with respect to $\bar{x}$ and then to z. The first step decomposes into P problems that can be solved at each node in parallel. For the pth node, x_{S_p} is updated as

$$x_{S_p}^{(p),k+1} = \underset{x_{S_p}}{\arg\min}\;\; f_p(x_{S_p}) + \sum_{l\in S_p}\Big((\lambda_l^{(p),k})^\top (x_l^{(p)} - z_l^k) + \frac{\rho}{2}\|x_l^{(p)} - z_l^k\|^2\Big)$$

$$= \underset{x_{S_p}}{\arg\min}\;\; f_p(x_{S_p}) + \sum_{l\in S_p}(\lambda_l^{(p),k} - \rho z_l^k)^\top x_l^{(p)} + \frac{\rho}{2}\|x_{S_p}\|^2\,.$$

After these updates, each component of z is updated as

$$
\begin{aligned}
z_l^{k+1} &= \arg\min_{z_l} \sum_{p\in\mathcal{V}_l} \Big({\lambda_l^{(p)}}^\top (x_l^{(p)} - z_l) + \frac{\rho}{2}\|x_l^{(p)} - z_l\|^2 \Big) \\
&= \arg\min_{z_l} \sum_{p\in\mathcal{V}_l} \Big({\lambda_l^{(p),k}}^\top x_l^{(p),k+1} - {\lambda_l^{(p),k}}^\top z_l + \frac{\rho}{2}\|x_l^{(p),k+1}\|^2 - \rho {z_l}^\top x_l^{(p),k+1} + \frac{\rho}{2}\|z_l\|^2 \Big) ,
\end{aligned}
$$

whose solution is given by equating its gradient to zero:

$$
\begin{aligned}
& \sum_{p\in\mathcal{V}_l} \Big(-\lambda_l^{(p),k} - \rho x_l^{(p),k+1} + \rho z_l^{k+1} \Big) = 0 \\
\iff \quad & \rho|\mathcal{V}_l| z_l^{k+1} = \sum_{p\in\mathcal{V}_l} \Big(\lambda_l^{(p),k} + \rho x_l^{(p),k+1} \Big) \\
\iff \quad & z_l^{k+1} = \frac{\sum_{p\in\mathcal{V}_l} \big(\lambda_l^{(p),k} + \rho x_l^{(p),k+1} \big)}{\rho|\mathcal{V}_l|} \\
\iff \quad & z_l^{k+1} = \frac{\frac{1}{\rho}\sum_{p\in\mathcal{V}_l} \lambda_l^{(p),k} + \sum_{p\in\mathcal{V}_l} x_l^{(p),k+1}}{|\mathcal{V}_l|} \, .
\end{aligned} \tag{5}
$$

Finally, each dual variable $\lambda_l^{(p)}$ is updated as

$$
\lambda_l^{(p),k+1} = \lambda_l^{(p),k} + \rho\big(x_l^{(p),k+1} - z_l^{k+1} \big) \, . \tag{6}
$$

Actually, the expression for z, (5) can be simplified if we notice that (6) implies

$$
\begin{aligned}
\sum_{p\in\mathcal{V}_l} \lambda_l^{(p),k+1} &= \sum_{p\in\mathcal{V}_l} \lambda_l^{(p),k} + \rho\big(\sum_{p\in\mathcal{V}_l} x_l^{(p),k+1} - \sum_{p\in\mathcal{V}_l} z_l^{k+1} \big) \\
&= \sum_{p\in\mathcal{V}_l} \lambda_l^{(p),k} + \rho \sum_{p\in\mathcal{V}_l} x_l^{(p),k+1} - \rho|\mathcal{V}_l| z_l^{k+1}
\end{aligned}
$$

and using (5)

$$
\begin{aligned}
&= \sum_{p\in\mathcal{V}_l} \lambda_l^{(p),k} + \rho \sum_{p\in\mathcal{V}_l} x_l^{(p),k+1} - \sum_{p\in\mathcal{V}_l} \lambda_l^{(p),k} - \rho \sum_{p\in\mathcal{V}_l} x_l^{(p),k+1} \\
&= 0 \, .
\end{aligned}
$$

Hence, the update for each z_l becomes the simple average over the copies spread out through $\mathcal{V}_l$:

$$
z_l^{k+1} = \frac{\sum_{p\in\mathcal{V}_l} x_l^{(p),k+1}}{|\mathcal{V}_l|} \, . \tag{7}
$$

It is precisely this update that makes the algorithm efficient only in the scenario where every induced subgraph is a star. Otherwise, computing the average (7) would require a consensus algorithm, or other techniques that would result in an increase of the number of communications.

The resulting algorithm is in Algorithm 1. Note that after each k iteration there was information flowing both ways in each edge just once. Therefore, each iteration of the algorithm requires one communication step.

Algorithm 1 2-block ADMM [1]

Initialization: set each $x_l^{(p),1}$, z_l^1, and $\lambda_l^{(p),1}$ with arbitrary values; choose $\rho > 0$; set $k = 1$

repeat

  for all $p = 1, \ldots, P$ [in parallel] **do**

    Update all its copies $x_{S_p}^{(p)} := \{x_l^{(p)}\}_{l \in S_p}$ with

$$x_{S_p}^{(p),k+1} = \arg\min_{x_{S_p}} \; f_p(x_{S_p}) + \sum_{l \in S_p} (\lambda_l^{(p),k} - \rho z_l^k)^\top x_l^{(p)} + \frac{\rho}{2} \|x_{S_p}\|^2$$

    Send $x_l^{(p),k+1}$ 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 z_l as

$$z_l^{k+1} = \frac{\sum_{p \in \mathcal{V}_l} x_l^{(p),k+1}}{|\mathcal{V}_l|}$$

    Send z_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

References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating method of multipliers*, Found. Trends Mach. Learn. **3** (2010), no. 1.