The 'D-Subspace' Algorithm for Online Learning over Distributed Networks

Yitong Chen⋆, *Student Member, IEEE*, Danqi Jin⋆, Jie Chen, *Senior Member, IEEE*

Notation. Normal font x, boldface small letters $\boldsymbol{x}$ and capital letters $\boldsymbol{X}$ denote scalars, column vectors and matrices, respectively. The notation $[\,\cdot\,]_{(:,j)}$ denote the j-th column. The superscript $(\cdot)^\top$ denotes the transpose operator. The mathematical expectation is denoted by $\mathbb{E}\{\cdot\}$. The set $\mathcal{N}_k$ denotes the neighbors of node k (including k itself), and $|\mathcal{N}_k|$ denotes its cardinality. Notation $\big[\boldsymbol{w}_\ell\big]_{\ell\in\mathcal{N}_k}$ denotes a matrix consisting of all $\boldsymbol{w}_\ell$ with $\ell\in\mathcal{N}_k$.

This material introduces the D-Subspace algorithm derived on the basis of the centralized algorithm [1], which originally addresses parameter estimation problems under a subspace constraint.

Consider a connected network with N agents. The set of all agents is denoted as $\mathcal{N}\triangleq\{1,2,\cdots,N\}$. Each agent $k\in\mathcal{N}$ is endowed with a strongly convex, real-valued and differentiable cost function $J_k(\boldsymbol{w}_k)$, which corresponds to the expectation of a loss function $G_k(\boldsymbol{w}_k;\boldsymbol{s}_{k,n})$:

$$J_k(\boldsymbol{w}_k)\triangleq\mathbb{E}\{G_k(\boldsymbol{w}_k;\boldsymbol{s}_{k,n})\},\tag{1}$$

where the expectation operator $\mathbb{E}\{\cdot\}$ is evaluated over the distribution of random data $\boldsymbol{s}_{k,n}$, with subscripts k and n representing node index and time instant, respectively. We denote the real-valued parameter vector $\boldsymbol{w}_k^\star\in\mathbb{R}^L$ as the unique minimizer of $J_k(\boldsymbol{w}_k)$. Define a matrix $\boldsymbol{W}^\star$ as:

$$\boldsymbol{W}^\star\triangleq[\boldsymbol{w}_1^\star,\ \ \boldsymbol{w}_2^\star,\ \ \cdots,\ \ \boldsymbol{w}_N^\star]\in\mathbb{R}^{L\times N}\tag{2}$$

The aim of this material is to explore a situation where $\boldsymbol{W}^\star$ is a low-rank matrix, with its rank being $r^\star$. In this case, we have:

$$\boldsymbol{w}_k^\star=\sum_{i=1}^{r^\star}\alpha_{k,i}^o\boldsymbol{c}_i=\boldsymbol{C}\cdot\boldsymbol{\alpha}_k^o\tag{3}$$

where $\{\boldsymbol{c}_i\}_{i=1}^{r^\star}$ are a set of basis, $\{\alpha_{k,i}^o\}_{i=1}^{r^\star}$ are corresponding weights, matrix $\boldsymbol{C}\triangleq[\boldsymbol{c}_1\,\boldsymbol{c}_2\cdots\boldsymbol{c}_{r^\star}]\in\mathbb{R}^{L\times r^\star}$, and vector $\boldsymbol{\alpha}_k^o\triangleq[\alpha_{k,1}^o\,\alpha_{k,2}^o\cdots\alpha_{k,r^\star}^o]^\top$. In this material, we assume that $\boldsymbol{\alpha}_k^o$ is known priorly. Substituting (3) into (2), we have:

$$\boldsymbol{W}^\star=\boldsymbol{C}\cdot\boldsymbol{\Theta}^o\tag{4}$$

where matrix $\boldsymbol{\Theta}^o\triangleq[\boldsymbol{\alpha}_1^o\,\boldsymbol{\alpha}_2^o\cdots\boldsymbol{\alpha}_N^o]\in\mathbb{R}^{r^\star\times N}$ is assumed to be known. Consequently, a centralized optimization problem emerges:

$$\begin{aligned}&\underset{\boldsymbol{w}_\ell:\ell\in\mathcal{N}}{\operatorname{argmin}}\sum\nolimits_{\ell=1}^{N}J_\ell(\boldsymbol{w}_\ell)\\&\text{s.t.}\quad\big[\boldsymbol{W}^\top\big]_{(:,j)}\in\mathcal{R}\Big([\boldsymbol{\Theta}^o]^\top\Big),\ \forall\,j\end{aligned}\tag{5}$$

where $\boldsymbol{W}\triangleq\big[\boldsymbol{w}_\ell\big]_{\ell\in\mathcal{N}}$ is an estimate of $\boldsymbol{W}^\star$, and $\mathcal{R}(\cdot)$ denotes the range space operator. In order to solve (5) iteratively, the gradient projection method can be applied, resulting in:

$$\begin{cases}\boldsymbol{\psi}_{k,n+1}=\boldsymbol{w}_{k,n}-\mu_k\nabla_{\boldsymbol{w}_k}G_k(\boldsymbol{w}_{k,n};\boldsymbol{s}_{k,n}) & (6)\\ \boldsymbol{\Psi}_{n+1}\triangleq[\boldsymbol{\psi}_{1,n+1},\,\boldsymbol{\psi}_{2,n+1},\,\cdots,\boldsymbol{\psi}_{N,n+1}] & (7)\\ \boldsymbol{\Phi}_{n+1}=\big[\mathcal{P}_{[\boldsymbol{\Theta}^o]^\top}\cdot(\boldsymbol{\Psi}_{n+1}^\top)\big]^\top=\boldsymbol{\Psi}_{n+1}\cdot\mathcal{P}_{[\boldsymbol{\Theta}^o]^\top} & (8)\end{cases}$$

where the projection matrix $\mathcal{P}_{[\boldsymbol{\Theta}^o]^\top}$ is defined as:

$$\mathcal{P}_{[\boldsymbol{\Theta}^o]^\top}\triangleq[\boldsymbol{\Theta}^o]^\top(\boldsymbol{\Theta}^o[\boldsymbol{\Theta}^o]^\top)^{-1}\boldsymbol{\Theta}^o.\tag{9}$$

Equations (6) – (8) are centralized solution, abbreviated as 'C-Subspace' in this material.

We would also like to pursue a distributed solution. Due to the fact that the network is connected and only local data exchanges are permitted in distributed processing, for each node k, we define a local optimal matrix $\boldsymbol{W}_k^\star$ as:

$$\boldsymbol{W}_k^\star\triangleq\big[\boldsymbol{w}_\ell^\star\big]_{\ell\in\mathcal{N}_k}\in\mathbb{R}^{L\times|\mathcal{N}_k|}\tag{10}$$

To ensure the uniqueness of $\boldsymbol{W}_k^\star$, we arrange its columns $\boldsymbol{w}_\ell^\star$ with $\ell\in\mathcal{N}_k$ in ascending order with respect to ℓ, such that $\boldsymbol{w}_\ell^\star$ is its $i_\ell^{(k)}$-th column. Within the neighborhood $\mathcal{N}_k$ of each node k, we have [1]:

$$\boldsymbol{w}_\ell^\star=\sum_{i=1}^{r_k^\star}\alpha_{\ell,i}^{(k)}\boldsymbol{c}_{k,i}=\boldsymbol{C}_k\cdot\boldsymbol{\alpha}_\ell^{(k)},\ \forall\ell\in\mathcal{N}_k\tag{11}$$

where $r_k^\star$ is the rank of matrix $\boldsymbol{W}_k^\star$, $\{\boldsymbol{c}_{k,i}\}_{i=1}^{r_k^\star}$ are a set of basis, with $\big\{\alpha_{\ell,i}^{(k)}\big\}_{i=1}^{r_k^\star}$ being corresponding weights with respect to node k, matrix $\boldsymbol{C}_k\triangleq[\boldsymbol{c}_{k,1}\,\boldsymbol{c}_{k,2}\cdots\boldsymbol{c}_{k,r_k^\star}]\in\mathbb{R}^{L\times r_k^\star}$, and vector $\boldsymbol{\alpha}_\ell^{(k)}\triangleq\big[\alpha_{\ell,1}^{(k)}\,\alpha_{\ell,2}^{(k)}\cdots\alpha_{\ell,r_k^\star}^{(k)}\big]^\top$. To ensure the uniqueness of (11), we require that for all k, all $\ell\in\mathcal{N}_k$ and all $i\in\{1,2,\cdots,r_k^\star\}$, there exists a $j\in\{1,2,\cdots,r^\star\}$, such that:

$$\alpha_{\ell,i}^{(k)}=\alpha_{\ell,j}^o\ \text{ and }\ \boldsymbol{c}_{k,i}=\boldsymbol{c}_j.\tag{12}$$

Similarly, in this material, we assume that $\boldsymbol{\alpha}_\ell^{(k)}$ is known

Y. Chen and D. Jin are co-first authors with equal contributions to this material. Y. Chen and J. Chen are with Centre of Intelligent Acoustics and Immersive Communications at School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China (chenyitong@mail.nwpu.edu.cn). D. Jin is with School of Electronic Information, Wuhan University, Wuhan, China (danqijin@whu.edu.cn).

[1]Note that notation $(\cdot)_{\ell,\cdot}^{(k)}$ denotes a quantity related to node ℓ, which is evaluated at node k and provided by node k.

priorly. Substituting (11) into (10), we have:

$$\boldsymbol{W}_k^\star = \boldsymbol{C}_k \cdot \boldsymbol{\Theta}_k \tag{13}$$

where matrix $\boldsymbol{\Theta}_k \triangleq \big[\boldsymbol{\alpha}_\ell^{(k)}\big]_{\ell\in\mathcal{N}_k} \in \mathbb{R}^{r_k^\star \times |\mathcal{N}_k|}$ is known, with $\boldsymbol{\alpha}_\ell^{(k)}$ being its $i_\ell^{(k)}$-th column. Consequently, a distributed optimization problem emerges at each node k as:

$$\begin{aligned} &\underset{\boldsymbol{w}_\ell:\ell\in\mathcal{N}_k}{\operatorname{argmin}} \sum\nolimits_{\ell\in\mathcal{N}_k} J_\ell(\boldsymbol{w}_\ell) \\ &\text{s.t.} \quad \big[\boldsymbol{W}_k^\top\big]_{(:,j)} \in \mathcal{R}\Big([\boldsymbol{\Theta}_k]^\top\Big),\ \forall\, j \end{aligned} \tag{14}$$

where $\boldsymbol{W}_k$ is an estimate of $\boldsymbol{W}_k^\star$, with its $i_\ell^{(k)}$-th column being an estimate of $\boldsymbol{w}_\ell^\star$. In order to solve (14) iteratively, the gradient projection method is applied, and local counterparts corresponding to the same estimate are further combined [2], resulting in:

$$\begin{cases} \boldsymbol{\psi}_{k,n+1} = \boldsymbol{w}_{k,n} - \mu_k \nabla_{\boldsymbol{w}_k} G_k(\boldsymbol{w}_{k,n}; \boldsymbol{s}_{k,n}) & (15) \\ \boldsymbol{\Psi}_{k,n+1} \triangleq \big[\boldsymbol{\psi}_{\ell,n+1}\big]_{\ell\in\mathcal{N}_k} & (16) \\ \boldsymbol{\Phi}_{k,n+1} = \big[\mathcal{P}_{[\boldsymbol{\Theta}_k]^\top} \cdot (\boldsymbol{\Psi}_{k,n+1}^\top)\big]^\top = \boldsymbol{\Psi}_{k,n+1} \cdot \mathcal{P}_{[\boldsymbol{\Theta}_k]^\top} & (17) \\ \boldsymbol{\phi}_{k,n+1}^{(\ell)} \triangleq \big[\boldsymbol{\Phi}_{\ell,n+1}\big]_{(:,i_k^{(\ell)})} & (18) \\ \boldsymbol{w}_{k,n+1} = \sum\limits_{\ell\in\mathcal{N}_k} a_{\ell k} \boldsymbol{\phi}_{k,n+1}^{(\ell)} & (19) \end{cases}$$

where the projection matrix $\mathcal{P}_{[\boldsymbol{\Theta}_k]^\top}$ is defined as:

$$\mathcal{P}_{[\boldsymbol{\Theta}_k]^\top} \triangleq [\boldsymbol{\Theta}_k]^\top (\boldsymbol{\Theta}_k [\boldsymbol{\Theta}_k]^\top)^{-1} \boldsymbol{\Theta}_k, \tag{20}$$

and $a_{\ell k}$ are single-task combination coefficients satisfying:

$$\sum_{\ell\in\mathcal{N}_k} a_{\ell k} = 1,\ a_{\ell k} \geq 0,\ \text{and}\ a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k. \tag{21}$$

Moreover, in several cases, the diagonal loading technique can be incorporated into (20), resulting in:

$$\mathcal{P}_{[\boldsymbol{\Theta}_k]^\top} \triangleq [\boldsymbol{\Theta}_k]^\top (\boldsymbol{\Theta}_k [\boldsymbol{\Theta}_k]^\top + \eta \boldsymbol{I})^{-1} \boldsymbol{\Theta}_k, \tag{22}$$

where $\eta > 0$ is diagonal loading factor with a small value. Equations (15) – (19) are distributed solution, abbreviated as ‘D-Subspace’ in this material.

References

[1] R. Nassif, S. Vlaski, and A. H. Sayed, “Adaptation and learning over networks under subspace constraints–part I: Stability analysis,” *IEEE Trans. Signal Process.*, vol. 68, pp. 1346–1360, 2020.

[2] Y. Chen, D. Jin, J. Chen, C. Richard, W. Zhang, G. Huang and J. Chen, “Online parameter estimation over distributed multitask networks with a rank-one model,” *32nd Eur. Signal Process. Conf.*, pp. 1042–1046, 2024.