Distributed Estimation of Common Signal Components across the Nodes of a Sensor Network

Charles Hovine and Alexander Bertrand *{charles.hovine, alexander.bertrand}@esat.kuleuven.be*

Contribution

We describe a scalable distributed algorithm for computing the solutions of the so-called MAXVAR canonical correlation analysis problem in a tree-topology network. Those solutions provide a basis to the subspace of jointly observed signal components.

Centralized Problem

- K nodes observe the M_k-channel signals $\boldsymbol{x}_k$.
- Find $\boldsymbol{W}_k$'s $\in \mathbb{C}^{Q\times M_k}$ and $\boldsymbol{s}$, a Q-dimensional random signal, minimizing

$$J(\underbrace{[\boldsymbol{W}_1^T,\dots,\boldsymbol{W}_K^T]^T}_{\boldsymbol{W}},\boldsymbol{s}) = \sum_{k=1}^{K}\mathbb{E}\left\{\left\|\boldsymbol{s}-\boldsymbol{W}_k^H\boldsymbol{x}_k\right\|^2\right\} \tag{1}$$

- $\boldsymbol{s}^* = \frac{1}{K}\boldsymbol{W}^H\boldsymbol{x}$ is a basis to the Q-dimensional signal subspace most adequatly describing cross-covariance between the nodes.
- $\boldsymbol{W}^*$ of (1) can be shown to satisfy the generalized eigenvalue decomposition:

$$\boldsymbol{R}_{D_{xx}}\boldsymbol{W}^* = \frac{1}{K^2}\boldsymbol{R}_{\boldsymbol{xx}}\boldsymbol{W}^*\boldsymbol{\Lambda} \tag{2}$$

- $\boldsymbol{W}^*$ is also solution to the optimization problem:

$$\min_{\boldsymbol{W}} \quad \mathrm{Tr}\left(\boldsymbol{W}^H\boldsymbol{R}_{D_{xx}}\boldsymbol{W}\right) \tag{3a}$$
$$\text{s.t.} \quad \boldsymbol{W}^H\boldsymbol{R}_{\boldsymbol{xx}}\boldsymbol{W} = K^2\boldsymbol{I}_Q \tag{3b}$$

where $\boldsymbol{R}_{D_{xx}} = \mathrm{Blkdiag}(\boldsymbol{R}_{\boldsymbol{x}_1\boldsymbol{x}_1},\dots,\boldsymbol{R}_{\boldsymbol{x}_K\boldsymbol{x}_K})$.

- Minimizing (1) is commonly referred to as the MAXVAR problem.

Distributed Algorithm

- Each node k share its observations $\boldsymbol{x}_k$ compressed by some matrix $\boldsymbol{C}_k^H \in \mathbb{C}^{Q\times M_k}$ with some node q.
- Node q cannot solve (3) without the raw observations, instead it solves

$$\min_{\tilde{\boldsymbol{W}}} \quad \mathrm{Tr}\left(\tilde{\boldsymbol{W}}^H\boldsymbol{R}_{D_{\overline{\boldsymbol{x}}_q\overline{\boldsymbol{x}}_q}}\tilde{\boldsymbol{W}}\right) \tag{4a}$$
$$\text{s.t.} \quad \tilde{\boldsymbol{W}}^H\boldsymbol{R}_{\overline{\boldsymbol{x}}_q\overline{\boldsymbol{x}}_q}\tilde{\boldsymbol{W}} = K^2\boldsymbol{I}_Q \tag{4b}$$

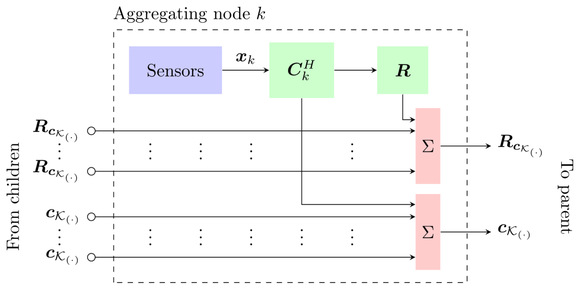
with $\overline{\boldsymbol{x}}_q = [(\boldsymbol{C}_1^H\boldsymbol{x}_1)^T \ \cdots \ \boldsymbol{x}_q^T \ \cdots \ (\boldsymbol{C}_K^H\boldsymbol{x}_K)^T]^T$

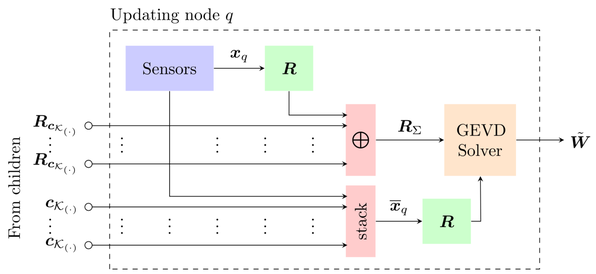
- Problem (4) is equivalent to (3) with the additional constraints $\boldsymbol{W}_k \in \mathcal{C}(\boldsymbol{C}_k) \quad \forall k \neq q$.
- Each node solves this problem in turns, updating its compression matrix as $\boldsymbol{C}_k^{i+1} = \boldsymbol{C}_k^i\tilde{\boldsymbol{W}}_k$.
- Solution of each iteration is also in the constraint set of the next iteration $\rightarrow$ monotonic increase of the objective and $\lim_{i\to\infty}\boldsymbol{C}_k^i = \boldsymbol{W}_k^*$.

Tree-Topology Networks

- We split the nodes in disjoint sets $\mathcal{K}_{(\cdot)}$ whose compressed observations will be summed and transmitted together.
- This is equivalent to (3) with additional constraints $\boldsymbol{W}_{\mathcal{K}_{(\cdot)}} \in \mathcal{C}(\boldsymbol{C}_{\mathcal{K}_{(\cdot)}})$ where $\boldsymbol{W}_{\mathcal{K}_{(\cdot)}}$ and $\boldsymbol{C}_{\mathcal{K}_{(\cdot)}}$ contain the corresponding stacked matrices of nodes in $\mathcal{K}_{(\cdot)}$.

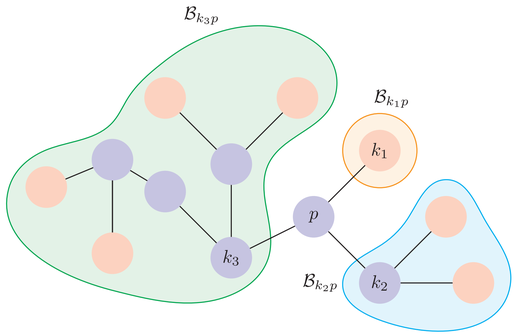
Node Operation in Tree-Topology Networks





The nodes operate in two states: *Updating* when solving the local problem, *Aggregating* when relaying compressed observations to the updating node.

Illustrative Example in a Tree-Topology Network



Node p is the current updating node. We distinguish three steps during iteration i:

I. *Aggregation:*

1. k_1 sends its compressed observations $\boldsymbol{C}_{k_1}^{iH}\boldsymbol{x}_{k_1}$ to p along with the covariance matrix of its compressed variables $\boldsymbol{C}_{k_1}^{iH}\boldsymbol{R}_{\boldsymbol{x}_{k_1}\boldsymbol{x}_{k_1}}\boldsymbol{C}_{k_1}^i$
2. k_2 collects and sums the observations of the nodes in $\mathcal{B}_{k_2p}$ and sends $\sum_{k\in\mathcal{B}_{k_2p}}\boldsymbol{C}_k^{iH}\boldsymbol{x}_k$ to p along with the sum of associated compressed variables covariance matrices
3. k_3 recursively collects and sums the observations of the nodes in $\mathcal{B}_{k_3p}$ and sends $\sum_{k\in\mathcal{B}_{k_3p}}\boldsymbol{C}_k^{iH}\boldsymbol{x}_k$ to p along with the sum of associated compressed variables covariance matrices

II. *Local solution:*

4. p computes the solutions $\tilde{\boldsymbol{W}}^H = [\boldsymbol{W}_p^H \ \tilde{\boldsymbol{W}}_{k_1}^H \ \tilde{\boldsymbol{W}}_{k_2}^H \ \tilde{\boldsymbol{W}}_{k_3}^H]$ of (5) and sends them to the corresponding subtrees.

III. *Update:*

5. p updates its compression matrix as $\boldsymbol{C}_k^{i+1} = \tilde{\boldsymbol{W}}_p$
6. k_1 updates its compression matrix as $\boldsymbol{C}_k^{i+1} = \boldsymbol{C}_k^i\tilde{\boldsymbol{W}}_{k_1}$
7. Node in subtree $\mathcal{B}_{k_2p}$ update their compression matrices as $\boldsymbol{C}_k^{i+1} = \boldsymbol{C}_k^i\tilde{\boldsymbol{W}}_{k_2}$
8. Node in subtree $\mathcal{B}_{k_3p}$ update their compression matrices as $\boldsymbol{C}_k^{i+1} = \boldsymbol{C}_k^i\tilde{\boldsymbol{W}}_{k_3}$

Distributed Algorithm (cont.)

- The nodes solve the local problem in turns:

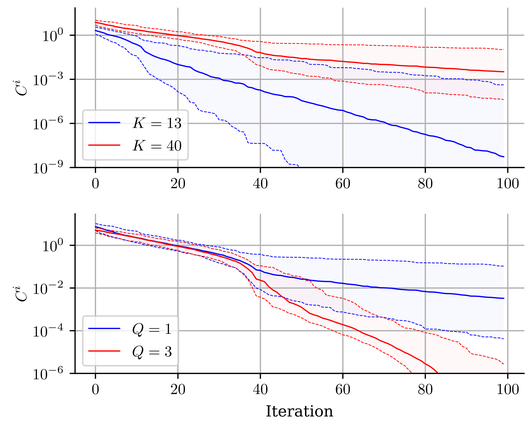
$$\min_{\tilde{\boldsymbol{W}}} \quad \mathrm{Tr}\left(\tilde{\boldsymbol{W}}^H\boldsymbol{R}_\Sigma\tilde{\boldsymbol{W}}\right) \tag{5a}$$
$$\text{s.t.} \quad \tilde{\boldsymbol{W}}^H\boldsymbol{R}_{\overline{\boldsymbol{x}}_q\overline{\boldsymbol{x}}_q}\tilde{\boldsymbol{W}} = K^2\boldsymbol{I}_Q \tag{5b}$$

where $\boldsymbol{R}_\Sigma = \mathrm{Blkdiag}(\boldsymbol{R}_{\boldsymbol{x}_q\boldsymbol{x}_q},\boldsymbol{R}_{\boldsymbol{c}_{\mathcal{K}_1}},\dots,\boldsymbol{R}_{\boldsymbol{c}_{\mathcal{K}_n}})$ and $\overline{\boldsymbol{x}}_q = [\boldsymbol{x}_q^T \ \boldsymbol{c}_{\mathcal{K}_1}^T \ \cdots \ \boldsymbol{c}_{\mathcal{K}_n}^T]^T$ with $\boldsymbol{c}_{\mathcal{K}} = \sum_{k\in\mathcal{K}}\boldsymbol{C}_k^H\boldsymbol{x}_k$

- $\boldsymbol{c}_{\mathcal{K}}$ and $\boldsymbol{R}_\Sigma$ can be computed through in-network summing.
- The sets are chosen as the subtrees which would be disconnected would the link between q and one of its neighbors be removed.
- There is a single update matrix $\tilde{\boldsymbol{W}}_{(\cdot)}$ per subtree.
- The current esitmate of $\boldsymbol{s}^*$ can be obtained as

$$\hat{\boldsymbol{s}}^i = \frac{1}{K}\sum\boldsymbol{C}_k^{iH}\boldsymbol{x}^k \tag{6}$$

Performance



- Validated with Monte-Carlo (MC) simulations in a tree-topology network with a branching factor of 3 and $M_k = 6$. Depicted cost is $C^i = 1 - \frac{J(\boldsymbol{W}^i)}{J(\boldsymbol{W}^*)}$.
- **Complexity** of local problems at node k is $\mathcal{O}\left((M_k + Q|\mathcal{N}_k|)^3\right)$ ($|\mathcal{N}_k|$: number of neighbors) and
- Average **communication cost** over each link is $\mathcal{O}((Q+T)Q)$ (T: window length).