Research Summary

Jiawei Sun

Electrical and Computer Engineering
University of Michigan

September, 2019

Outline

1. Differential Microphones Arrays based on Differential Equation
   - Linear DMA
   - Circular DMA
2. Distributed Algorithms of PCA
   - Backgrounding
   - Average Consensus Algorithm
   - Distributed PCA

Outline

1. Differential Microphones Arrays based on Differential Equation
   - Linear DMA
   - Circular DMA

2. Distributed Algorithms of PCA
   - Backgrounding
   - Average Consensus Algorithm
   - Distributed PCA

- Why we need Distributed Algorithms of PCA (D-PCA)?
  1. data are collected/stored in a distributed network
  2. memory limitation
  3. privacy issue
  4. parallel clusters
- How D-PCA work for parallel processors?
  1. each node calculates its local value of PCA
  2. communicate with its neighbor nodes
  3. update with a weighted average of its neighbors values
- Application
  1. classify word documents
  2. array processing

- The designs of D-PCA algorithms differ in how data are divided in the network:
  1. Distributed columns observations (DCO)
  2. Distributed rows observations (DRO)

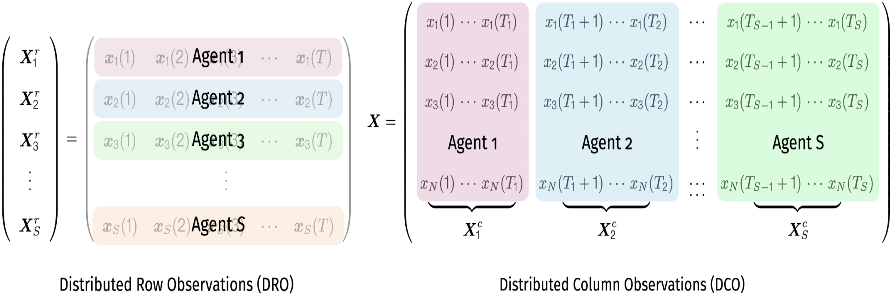

Figure: Data Model

- The DCO setting assumes that each agent observes a subset of columns of $X \in C^{N*T}$:

$$X = (X_1^c, X_2^c, ..., X_S^c)$$

where $X_i^c \in C^{N*Ti}$ is the column-partitioned sub-matrix and $\sum_{i=1}^{S} T_i = T$.

- DCO applies when high-dimension data are stored in different sites in a network.

- The DRO setting assumes that each agent observes only a subset of rows of $X \in C^{N*T}$:

$$X = ((X_1^r)^T, (X_2^r)^T, ..., (X_S^r)^T)^T$$

  where $X_i^r \in C^{Ni*T}$ is the column-partitioned sub-matrix and $\sum_{i=1}^{S} N_i = N$.
- DRO applies when data have a multidimensional time series and each sample is distributed across the nodes.

Backgrounding

Two Types of Communication Model

- The designs of D-PCA algorithms also differ in the types of communication among each node:
  1. master-slave type
  2. mesh type

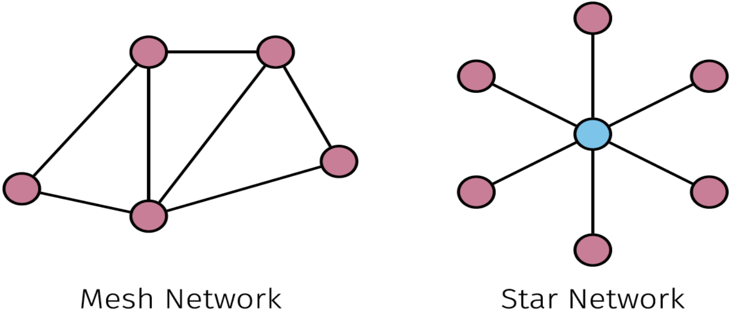

Figure: Communication Model

Two Types of Communication Model

- How master-slave model work?
  1. in local stage, each slave node solves a local PCA
  2. send local PCA results to the master node
  3. in global stage, the master node computes the global PCA from the aggregated data
- How mesh model work?
  1. all nodes and links perform the same function
  2. all nodes exchange partial computations
  3. transmitting information from one node to another may require multihop communications

Outline

1. Differential Microphones Arrays based on Differential Equation
   - Linear DMA
   - Circular DMA
2. Distributed Algorithms of PCA
   - Backgrounding
   - Average Consensus Algorithm
   - Distributed PCA

Average Consensus Algorithm

- Why need Average Consensus(AC) Algorithm?
  In D-PCA algorithm, the key is to aggregate and share information across nodes.
  - For master-slave model, we can centralize information in master node.
  - For mesh model, this has to be done with a sequence of computation steps adaptable to the network structure.

  We cannot centralize information directly in mesh model, so we take the iterative method to aggregate data

Average Consensus Algorithm

- Assume that the system of N sensor nodes is connected through a communication network. It is modeled by a graph whose topology is represented by the corresponding Laplacian matrix L.[4]
- The elements of matrix L [5]

$$l_{ij} = \begin{cases} d_j, \ i = j \\ -1, \ i \text{ communicates with j} \\ 0, \ \text{else} \end{cases}$$

where d_j is the number of its neighbor.

- Let $W = I - \varepsilon L$. The following linear iterative algorithm :

$$x_j(t+1) = W_{jj}x_j(t) + \sum_{k \in N} W_{jk}x_k(t)$$

$$x(t+1) = Wx(t)$$

Average Consensus Algorithm

- $W1 = 1$, so the eigenvector is 1 and eigenvalue is 1. The second largest eigenvalue of W, $\lambda_2 < 1$.
- No matter what the initial node values are, we must have

$$\lim_{t\to\infty} x(t) = \lim_{t\to\infty} W^t x(0) = \frac{1}{n}11^T x(0)$$

- All elements in x(t) are the same, and are the average of x(0) elements.
- Therefore, by AC algorithm, each node only need to communicates with its neighbor nodes. After iterationwe can compute the average of all nodes.

Outline

1. Differential Microphones Arrays based on Differential Equation
   - Linear DMA
   - Circular DMA

2. Distributed Algorithms of PCA
   - Backgrounding
   - Average Consensus Algorithm
   - Distributed PCA

- A distributed PCA method can be obtained by simply approximating the global correlation matrix via the AC subroutine,

$$\hat{R}_{u,i} = N \cdot AC(\{u_i u_i^T\}_{i=1}^N; L) \approx R_u \tag{31}$$

- In other words, each agent obtains an approximate of the global correlation matrix and the desired PCA can be then computed from $\hat{R}_{u,i}$.

- Eigenvalue decomposition of R_x and reduce its dimension to p-dim.

$$R_u = \sum_{i=1}^{N} \lambda_i u_i u_i^T \xrightarrow{\text{reduce dim}} R_u \approx \sum_{i=1}^{P} \lambda_i u_i u_i^T \tag{32}$$

- Supposed that we have N distributed nodes, so the optimization problem is

$$\begin{aligned} &\min \sum_{i \in V} -x_i^T R_u x_i \\ &s.t.\ x_i^T x_i = 1,\ i \in V \\ &x_i = x_j,\ \forall (i,j) \in E \end{aligned} \tag{33}$$

Distributed PCA

PDMM for DCO

- The PDMM [7] solves problem in this form:

$$\min \sum_{i \in V} f_i(x) \quad s.t. \; A_{ij}x_i + A_{ij}x_j = c_{ij}, \; \forall (i,j) \in E \tag{34}$$

where

$$f_i(x) = -u_i^T R_x u_i \tag{35}$$

$$\begin{cases} A_{ij} = I, \; i < j \\ A_{ji} = -I, \; \text{others} \end{cases} \tag{36}$$

$$c_{ij} = 0 \tag{37}$$

- We denote δ as the Lagrangian multiplier, and the Lagrangian of this primal problem can be constructed as

$$L_p(x,\delta) = \sum_{(i,j)\in E} \delta_{ij}^T (c_{ij} - A_{ij}x_i - A_{ji}x_j) + \sum_{i\in V}\left[f_i(x_i) + \theta_i^T(1 - x_i^T x_i)\right] \tag{38}$$

- The Augmented Primal-Dual Lagrangian function is

$$L_P = \sum_{i\in V}\left[f_i(x_i) - \sum_{j\in N(i)} \lambda_{j|i}^T (A_{ij}x_i - c_{ij}) - f_i^*(A_i^T\lambda_i)\right] + h(x_i,\lambda_i) \tag{39}$$

where

$$h(x_i,\lambda_i) = \sum_{(i,j)\in E} \left(\frac{1}{2}\|A_{ij}x_i + A_{ji}x_j + c_{ij}\|^2 - \frac{1}{2}\left\|\lambda_{i|j} - \lambda_{j|i}\right\|^2\right) \tag{40}$$

- At iteration k, the update scheme of PDMM is

$$\begin{aligned} x_i^{k+1} &= x_i^k - \alpha \nabla_{x_i} L_P \\ \theta_i^{k+1} &= \theta_i^k + \alpha \nabla_{\theta_i} L_P \\ \lambda_{i|j}^{k+1} &= \lambda_{i|j}^k + (c_{ij} - A_{ji} x_j^k - A_{ij} x_i^k), \ \forall i \in V, \ j \in N(i) \end{aligned} \tag{41}$$

where

$$\nabla_{x_i} L_P = -2R_{u_i} x_i - \sum_{j \in N(i)} \lambda_{j|i}^T A_{ij} - 2\theta_i x_i + \sum_{(i,j) \in E} A_{ij}(A_{ij} x_i + A_{ji} x_j) \tag{42}$$

$$\nabla_{\theta_i}(L_P) = 1 - 2x_i^T x_i \tag{43}$$

Algorithm 1 PDMM

1: Initialize as x_i^0, $\lambda_{i|j}^0$, θ_i^0 for all nodes
2: **for** $k = 1$ to K **do**
3: **for** $i = 1$ to N **do**
4:
$$x_i^{k+1} = x_i^k - \alpha \nabla_{x_i} L_P$$
$$\theta_i^{k+1} = \theta_i^k + \alpha \nabla_{\theta_i} L_P$$
$$\lambda_{i|j}^{k+1} = \lambda_{i|j}^k + (c_{ij} - A_{ji} x_j^k - A_{ij} x_i^k), \ \forall i \in V, \ j \in N(i)$$
5: **end for**
6: **end for**

- We introduce Rayleigh quotient to replace the constrain $x_i^T x_i = 1$, and the optimization problem is

$$\min \sum_{i \in V} \frac{-x_i^T R_u x_i}{x_i^T x_i} \qquad (44)$$
$$s.t.\ x_i = x_j,\ \forall (i,j) \in E$$

Algorithm 2 PDMM (Rayleigh Quotient)

Initialize as x_i^0, $\lambda_{i|j}^0$ for all nodes
for $k = 1$ to K **do**
  for $i = 1$ to N **do**

$$x_i^{k+1} = x_i^k - \alpha \nabla_{x_i} L_P$$

$$\lambda_{i|j}^{k+1} = \lambda_{i|j}^k + (c_{ij} - A_{ji}x_j^k - A_{ij}x_i^k), \ \forall i \in V, \ j \in N(i)$$

  end for
end for

- $$\begin{aligned} &\min \sum_{i \in V} -x_i^T R_{u_i} x_i \\ &s.t.\ A_{ij} x_i + A_{ij} x_j = c_{ij},\ \forall (i,j) \in E \end{aligned} \tag{45}$$

  where at iteration k

  $$\begin{cases} A_{ij} = I,\ i < j \\ A_{ij} = -I,\ i > j \\ A_{ij} = \begin{pmatrix} x_1^{k-1} & \cdots & x_N^{k-1} \end{pmatrix},\ i = j \end{cases} \tag{46}$$

Algorithm 3 PDMM (Time-varing constrains)

1: Initialize as x_i^0, $\lambda_{i|j}^0$, A_{ii} for all nodes
2: **for** $k = 1$ to K **do**
3: **for** $i = 1$ to N **do**
4:

$$x_i^{k+1} = x_i^k - \alpha \nabla_{x_i} L_P$$

$$\lambda_{i|j}^{k+1} = \lambda_{i|j}^k + (c_{ij} - A_{ji} x_j^k - A_{ij} x_i^k), \ \forall i \in V, \ j \in N(i)$$

5: **end for**

$$A_{ii} = \begin{pmatrix} x_1^{k-1} & \cdots & x_N^{k-1} \end{pmatrix}$$

6: **end for**

For Further Reading I

- Benesty, Jacob, and Chen Jingdong. Study and design of differential microphone arrays. Vol. 6. Springer Science Business Media, 2012.
- Benesty, Jacob, Jingdong Chen, and Israel Cohen. Design of Circular Differential Microphone Arrays. Vol. 12. Switzerland: Springer, 2015.
- Chatelin, Franoise, ed. Eigenvalues of Matrices: Revised Edition. Society for Industrial and Applied Mathematics, 2012.
- Wu, Sissi Xiaoxiao, et al. "A Review of Distributed Algorithms for Principal Component Analysis." Proceedings of the IEEE 106.8 (2018): 1321-1340.
- Scaglione, Anna, Roberto Pagliari, and Hamid Krim. "The decentralized estimation of the sample covariance." 2008 42nd Asilomar Conference on Signals, Systems and Computers. IEEE, 2008.

For Further Reading II

- Qu, Yongming, et al. "Principal component analysis for dimension reduction in massive distributed data sets." Proceedings of IEEE International Conference on Data Mining (ICDM). 2002.
- Zhang, Guoqiang, and Richard Heusdens. "Bi-alternating direction method of multipliers over graphs." 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2015.