Signal Processing and Optimization for Big Data

24/10/2025

1)Let us consider the LASSO problem

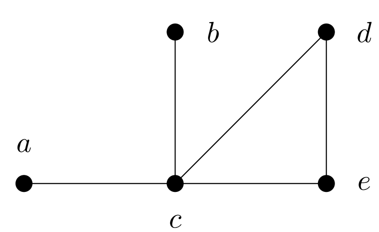
$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2 + \alpha\|\mathbf{x}\|_1, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{N\times M}$, $\mathbf{x} \in \mathbb{R}^{M\times 1}$ and $\alpha > 0$.

- Assuming to have a Fusion Center, write a Matlab/Python code to solve the problem in a distributed fashion by **splitting across the examples**.
- Compute the disagreement $D[k] = \sum_{i=1}^{N_a}\sum_{j=1}^{N_a} \|\mathbf{x}_i[k] - \mathbf{x}_j[k]\|^2$ and plot its temporal behaviour.
- Denoting with $\boldsymbol{x}_0$ the solution obtained through the centralized algorithm, compute and plot the temporal behavior of the MSE between the centralized and the distributed solution $E[k] = \|\mathbf{x}[k] - \mathbf{x_0}\|^2$.
- Test the developed code on a data-set of your choice. For instance, you can consider the regression task based on the *mpg-dataset*.
- Compare the obtained solution with respect to the one obtained with the centralized approach.

Hint: Use, for example, $N_a = 5$, $\alpha = 10$, $\rho = 10$. It could be useful to normalize the data.

2)Assuming to have the following network



- Implement a consensus algorithm to solve the distributed-LASSO problem considered in the previous exercise.
- Implement the consensus procedure based on the minimization of the Total Variation function, defined as

$$TV(\mathbf{x}) = \sum_{i=1}^{N_a}\sum_{j=1}^{N_a} a_{i,j}\|\mathbf{x}_i - \mathbf{x}_j\|^2, \tag{2}$$

with $\{a_{i,j}\}_{i,j=1}^{N_a}$ denoting the entries of the graph djacency matrix.

- Plot the temporal behavior of the disagreement between nodes.
- Plot the temporal behavior of the squared error between the centralized and the distributed method, and test the obtained solution with a real data-set.

3)Let us consider the (regularized) SVM problem

$$\min_{(\boldsymbol{\beta},\beta_0)} \frac{1}{N}\sum_{i=1}^{N} \max(0, 1 - y_i(\boldsymbol{\beta}^t \mathbf{x}_i + \boldsymbol{\beta}_0)) + \lambda\|\boldsymbol{\beta}\|_2 \tag{3}$$

- Write a Python/Matlab function to solve the problem in a centralized fashion.
- Write a Python/Matlab function to solve the problem in a fully-distributed fashion, assuming to deal with the network considered in the previous exercise.
- Test the function you developed in a real data-set of your choice. You can use, for instance, the *Breast Cancer dataset*, which can be loaded using the Matlab command `load cancer_dataset`.
- How does the problem change if we consider $\|\boldsymbol{\beta}\|_1$ as regularization term?