# Distributed Networked Learning with Correlated Data

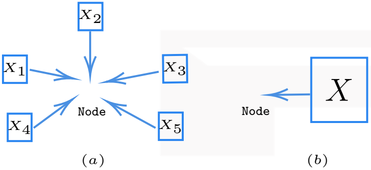
Lingzhou Hong, Alfredo Garcia, and Ceyhun Eksin

*Department of Industrial and Systems Engineering*
Texas A&M University, College Station, TX

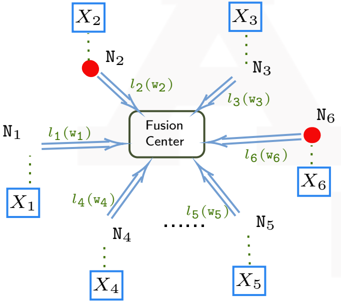
59th Conference on Decision and Control
December 14th-18th 2020

CDC 2020
Jeju Island, Korea

# Centralized vs Distributed Systems



Centralized Algorithms



Federated Learning Algorithms

Centralized system issues:

- security and privacy
- collection and transmission
- processing and storage

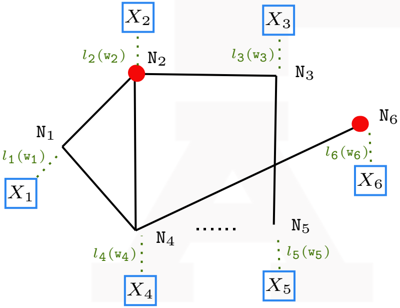
Isolated system issues:

- correlated and heteroscedastic local datasets
- disparities in performance

Federated Learning issues:

- security and privacy
- processing and storage

# Distributed algorithm with regularization



Distributed Networked Algorithms

Algorithm design consideration:

- global and local noise
- network of local learners
- asynchronous stochastic gradient descent
- general least squares problem
- weighted regularization penalty
- *Stochastic Gradient with Network regularization (SGN)*

# Centralized learning problem

- A large dataset $(\mathbf{X}, \mathbf{y})$ is divided into $N$ disjoint subsets, and can be expressed as $\mathbf{X} = [\mathbf{X}_i^\intercal, \ldots, \mathbf{X}_N^\intercal]^\intercal$ and $\mathbf{y} = [y_1^\intercal, \ldots, y_N^\intercal]^\intercal$.
- For each dataset, the model is:

$$\mathbf{y}_i = \mathbf{X}_i \mathbf{w}^* + \varepsilon_i + \Lambda_i \xi,$$

- the covariance matrix for $\mathbf{y}_i$ is

$$\Omega_i := \mathbb{E}(\varepsilon_i + \Lambda_i \xi)(\varepsilon_i + \Lambda_i \xi)^\intercal = \sigma_i^2 \mathbf{I} + \Lambda_i^2.$$

- The data model is:

$$\mathbf{y} = \mathbf{X}\mathbf{w}^* + \varepsilon + \Lambda \xi,$$

  where $\Lambda = [\Lambda_1, \ldots, \Lambda_N]^\intercal$ and $\varepsilon = [\varepsilon_i^\intercal, \ldots, \varepsilon_N^\intercal]^\intercal$.
- The covariance matrix for $\mathbf{y}$ is:

$$\Omega := \mathbb{E}(\varepsilon + \Lambda \xi)(\varepsilon + \Lambda \xi)^\intercal = \Sigma + \Lambda \Lambda^T.$$

- A centralized formulation of the generalized least squares (GLS):

$$\mathcal{L}_c \triangleq \frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^T \Omega^{-1} (\mathbf{y} - \mathbf{X}\mathbf{w}).$$

# Local learning problem

Each node minimizes a weighted loss function with a regularization term:

$$\min_{\mathbf{w}_i} f_i(\mathbf{w}_i) + \lambda \rho_i(\mathbf{w}_i)$$

- $f_i(\mathbf{w}_i)$ is a local loss function

$$f_i(\mathbf{w}_i) = \frac{1}{2}(\mathbf{y}_i - \mathbf{X}_i\mathbf{w}_i)^T \Omega_i^{-1} (\mathbf{y}_i - \mathbf{X}_i\mathbf{w}_i)$$

.

- $\rho_i(\mathbf{w}_i)$ is a weighted penalty function

$$\rho_i(\mathbf{w}_i) = \frac{1}{2} \sum_{j \neq i} \frac{a_{i,j}}{\mathrm{tr}(\Omega_j)} \left\| \mathbf{w}_i - \mathbf{w}_j \right\|^2$$

- $\lambda \geq 0$ is the regularization parameter.
- Local problem is equivalent to the global objective:

$$\mathcal{L}(\mathbf{w}_1, \ldots, \mathbf{w}_N) \triangleq \frac{1}{v} \sum_{i=1}^{N} \frac{1}{\mathrm{tr}(\Omega_i)} \left( f_i(\mathbf{w}_i) + \frac{\lambda}{2} \rho_i(\mathbf{w}_i) \right)$$

where $v = \sum_{i=1}^{N} 1/\mathrm{tr}(\Omega_i)$ is a normalization constant.

# SGN algorithm and convergence measurements

- SGN updates: for node $i$,

$$\mathbf{w}_{i,k+1} = \mathbf{w}_{i,k} - \Gamma(\nabla f_{i,k} + \lambda \nabla \rho_{i,k}),$$

  where $\Gamma$ is the stepsize.
- We embed the discrete time process in into a continuous-time domain.
- Weighted average solution at time $t$ is defined as:

$$\hat{\mathbf{w}}_t = \frac{1}{v} \sum_{i=1}^{N} \frac{\mathbf{w}_{i,t}}{\mathsf{tr}(\Omega_i)},$$

- To measure the the performance of SGN, we consider *regularity* measure $\bar{V}_t$ and *optimality* measure $U_t$.

# Regularity

Regularity Measure is defined as: $\bar{V}_t = \frac{1}{v}\sum_{i=1}^{N} \frac{\|\mathbf{w}_{i,t} - \hat{\mathbf{w}}_t\|^2}{2\mathrm{tr}(\Omega_i)}$

## Theorem

*Suppose $\|g_{i,t}\| \leq \eta$, for all $i$ and some $\eta > 0$, then*

$$\mathbb{E}[\bar{V}_t] \leq \frac{\gamma C_1}{2(\kappa + \lambda a_2)} + (\bar{V}_0 - \frac{\gamma C_1}{2(\kappa + \lambda a_2)})e^{-2(\kappa+\lambda a_2)\gamma t}$$

*In the long run,*

$$\lim_{t\to\infty} \mathbb{E}[\bar{V}_t] \leq \frac{\gamma C_1}{2(\kappa + \lambda a_2)}$$

- The upper bound of the $\mathbb{E}[\bar{V}_t]$ decreases as $\lambda$ and $a_2$ increases.
- The constant term $C_1$ is determined by data $\mathbf{X}$ and the matrices $\Lambda_i$. In particular, $C_1$ is small when we have nodes that are less affected by the noise.

# Optimality

Optimality Measure is defined as: $U_t = \frac{1}{2}\left\|\hat{\mathbf{w}}_t - \mathbf{w}^*\right\|^2$

## Theorem

$$\mathbb{E}[U_t] \leq e^{-2\kappa\gamma t}U_0 + \frac{\gamma}{2\kappa}\Big(\frac{\mu-\kappa}{\lambda a_2}C_1 + C_2\Big)\Big(1 - e^{-2\kappa\gamma t}\Big),$$

*in the long run,*

$$\lim_{t\to\infty}\mathbb{E}[U_t] \leq \frac{\gamma}{2\kappa}\left(\frac{\mu-\kappa}{\lambda a_2}C_1 + C_2\right)$$

- The penalty constant $\lambda$ and the algebraic connectivity $a_2$ reduce the bound on the expected consistency.
- The constant $C_2$ is determined by the data $\mathbf{X}$, weight matrices $\Lambda_i$'s, and covariance matrices $\Omega_i$'s. It suggests that we can only improve performance by the addition of new nodes that have access to more reliable data.

# Predicting head movement activities

- We consider a real-world problem of predicting the activity of individuals from head movement from GLEAM dataset.
- 18 predictors from the readings of the sensors.
- $N = 37$ computing nodes with a complete network structure.

| activities | eating | working |
|---|---|---|
| encode | $0$ | $1$ |
| percent-training | $20\%$ | $80\%$ |
| percent-testing | $10\%$ | $90\%$ |

| | training | testing |
|---|---|---|
| participants | 37 | 1 |
| data points | 96,829 | 2,617 |

# Approximations in practical implications

- The gradients are approximated with mini-batch of size of $m = 100$.
- The empirical covariance matrices are approximated with $50$ mini-batch samples.
- The covariance matrix is approximated by the trace of the current empirical covariance matrix with a fading memory update rule:

$$\mathsf{tr}(\Omega_{i,k+1}) = \varphi \mathsf{tr}(\Omega_{i,k}) + (1-\varphi)\mathsf{tr}(\hat{\Omega}_{i,k+1}),$$

where $\mathsf{tr}(\hat{\Omega}_{i,k+1})$ is the $i$-th covariance matrix and $\varphi \in (0,1)$.

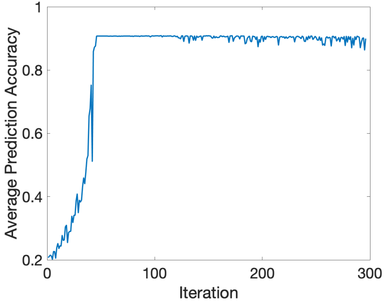
# Numerical settings

- The activity estimate of the $i$-th node is given by $\mathbf{X}_i\mathbf{w}_i$.
- At step $k$, we predict activity as $0$ if $\mathbf{X}_j\hat{\mathbf{w}}_k < 0.5$ and $1$ otherwise.
- The prediction accuracy of SGN is given by

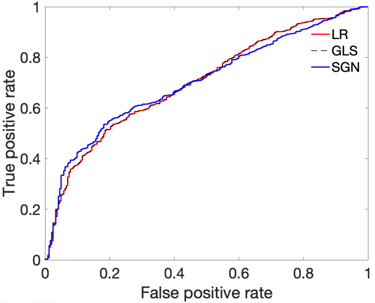
$$\text{accuracy}_k = 1 - \frac{\left\|\mathbf{y}_{\text{test}} - \mathbf{1}_{\{\mathbf{X}_{\text{test}}\hat{\mathbf{w}}_k > 0.5\}}\right\|^2}{m},$$

- The average prediction accuracy is defined as the mean prediction accuracy at step $k$ over $10$ runs.
- Parameter setting: $\varphi = 0.9$, $\Gamma = 300$, $\lambda = 100$, and $T = 300$.
- SGN is compared with GLS and logistic regression (LR).

# Numerical Results



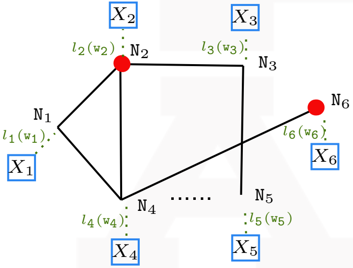
**(a)** SGN average prediction accuracy at each iterations.



**(b)** The ROC curve of LR($0.7014$), GLS ($0.7014$), and SGN ($0.7037$).

The average prediction accuracy of SGN ($0.8972$) is close to that of LR ($0.9079$) and GLS ($0.8991$).

# Summary



Distributed Networked Algorithms

- data new challenges
- distributed networked architecture for linear learning
- stochastic gradient updates based upon local samples
- network regularization penalizes high *local* variability
- continuous and asynchronous model averaging
- illustrating robustness