Distributed Linear Regression:

We have tried to solve distributed linear regression in two different scenarios. In the first case, several agents partially observe the entries of $\mathbf{X}_k$ and are reluctant to share their data for privacy reasons. These agents fit a model of their own by applying Recursive Least Square(RLS) algorithm on the partial training data set and then send their local estimates to the central coordinator. The central coordinator upon receiving the local estimates by all the agents, runs another RLS on these local estimates and calculates the global estimate. Mathematically, consider in a linear regression model where X_k and y_k evolve as per following state space equation:

$$X_{k+1} = AX_k + u_k$$
$$y_k = CX_k + v_k$$

where A and C are suitable matrices and $\{v_k\}_{k\geq 0}$ and $\{u_k\}_{k\geq 0}$ are i.i.d. zero mean unit variance Gaussian random variables. The regression co-efficients are unknown and goal is to predict it or estimate the y_k's. Now suppose we have agent $n = 1, ..., N$ and agent n observes the entries of $\mathbf{X}_k$ from a set $I_n \subset \{1, 2, ..., p\}$. Let $\mathbf{X}_k^n$ denotes the predictor variables observed by agent n. Agent n now runs a RLS based on observations $\{Y_k, \mathbf{X}_k^n\}$, say for K iterations. If RLS estimate of agent n at time k is $\hat{\beta}_k^n$, then it's prediction of Y_k will be:

$$\hat{Y_k}^n = (\hat{\beta}_k^n)^T \mathbf{X}_k^n$$

In order to improve the estimates without sharing the observations, the agents will transfer their local estimates to the central agent and the central agent in turn in turn forms an estimate

$$\hat{Y_k} = \textstyle\sum_{n=1toN} \hat{\omega}_k^n \hat{Y_k}^n$$

where the weights $\hat{\omega}_k^n$ are obtained by running another RLS algorithm based on observations $\{Y_k, \hat{Y}_k^1, \hat{Y}_k^2, ..., \hat{Y}_k^N\}$.

In the second scenario, we consider that we do not have a central coordinator. But instead we have a graph with agents as the nodes. An agent, after each time step, sends its estimate to its neighbors in the graph. The estimate at any agent is formed by a linear combination of its observations and the estimates obtained at previous times from its neighbors. The coefficients in the linear combination are adapted using RLS.

Let us consider a graph where $\aleph_n$ denotes the neighbouring set of node n , where $\aleph_n \subset \{1, 2, ..., N\}$. As in the first scenario, let $\mathbf{X}_k^n$ denotes the predictor variables observed by agent n. Now agent n performs local estimation using RLS on dataset comprising of $\{Y_k, \mathbf{X}_k^n\}$ as well as the previous step estimates received from it's neighbours, $\hat{Y}_{k-1}^{n'}$, where $n' \in \aleph_n$. So agent n forms an estimate

$$\hat{Y}_k^n = (\hat{\beta_k^n})^T \mathbf{X}_k^n + \sum_{\forall n' \in \aleph_n} \hat{\omega}_k^{n'} \hat{Y}_{k-1}^{n'}$$

where $\hat{\beta_k^n}$ and $\hat{\omega}_k^{n'}$ are obtained by running RLS algorithm based on observations $\{Y_k, \mathbf{X}_k^n, \{\hat{Y}_{k-1}^{n'}\}_{\forall n' \in \aleph_n}\}$. Since the estimates of different agents will vary depending upon the partial dataset available to them and also on estimates of their neighbours, the agents with more number of neighbours are expected to perform well, i.e. their estimates will be more close to the global estimate based on all training data and hence to the actual response. This scenario is simulated with a random graph consisting of 30 nodes and probability of existance of an edge between any two nodes is 0.2 and in the results, it is seen that accuracy of estimation increases with increase in number of neighbours.