Fully Decentralized Federated Learning

Anusha Lalitha
University of California San Diego
alalitha@eng.ucsd.edu

Shubhanshu Shekhar
University of California San Diego
shshekha@eng.ucsd.edu

Tara Javidi
University of California San Diego
tjavidi@eng.ucsd.edu

Farinaz Koushanfar
University of California San Diego
fkoushanfar@eng.ucsd.edu

Abstract

We consider the problem of training a machine learning model over a network of users in a fully decentralized framework. The users take a Bayesian-like approach via the introduction of a belief over the model parameter space. We propose a distributed learning algorithm in which users update their belief by aggregate information from their one-hop neighbors to learn a model that best fits the observations over the entire network. In addition, we also obtain sufficient conditions to ensure that the probability of error is small for every user in the network. Finally, we discuss approximations required for applying this algorithm for training Neural Networks.

1 Introduction

Mobile computing devices have seen a rapid increase in their computational power as well as storage capacity. Aided by this increased computational power and abundance of data, as well as due to privacy and security concerns, there is a growing trend towards training machine learning models over networks of such devices using only local training data. The field of *Federated learning* initiated in McMahan et al. (2017) and Konečnỳ et al. (2016) considers the problem of learning a centralized model based on private training data of a large number of users. More specifically, this framework is characterized by a huge number of decentralized users who are connected to a centralized server. The different users generate possibly non-iid data and furthermore it is assumed that communications between the users and the central server incur large costs. McMahan et al. (2017) proposed the *federated optimization* algorithm in which the central server randomly selects a fraction of the users in each round, shares the current global model with them, and then averages the updated models sent back to the server by the selected users.

In this paper, we also consider the problem of training a machine learning model over a network of devices which differs from the federated learning framework of McMahan et al. (2017) in the following ways:

- *Fully decentralized model*, we do not require the existence of a centralized controller. Instead, in our setting, users can only communicate with their one-hop neighbors.
- *Localized Inputs*, the training data available to an individual user is not sufficient to uniquely identify the underlying model. Thus the users must collaborate with each other to learn the optimal model.

Our problem formulation does away with the need of having a centralized controller, and instead aims to collaboratively learn the optimal model by local information exchange.

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

Contributions: Our contributions are as follows: 1) we first present a formal description of the fully decentralized federated learning problem; 2) we then present a distributed learning algorithm and obtain theoretical bounds on its performance; and 3) we describe the approximations required to employ this algorithm for training Deep Neural Networks (DNNs) in a decentralized manner.

Notation: We use boldface for vectors and denote the i-th element of vector $\mathbf{v}$ by v_i. We let $[n]$ denote $\{1, 2, \ldots, n\}$, $\mathcal{P}(A)$ the set of all probability distributions on a set A, $|A|$ denotes the number of elements in set A, and $D_{\mathrm{KL}}(P_Z||P'_Z)$ the Kullback–Leibler (KL) divergence between two probability distributions $P_Z, P'_Z \in \mathcal{P}(\mathcal{Z})$.

2 The Model

Consider a group of N individual users. Each user i has access to a dataset $\mathcal{D}_i$ containing n samples $\left\{(X_i^{(1)}, Y_i^{(1)}), (X_i^{(2)}, Y_i^{(2)}), \ldots, (X_i^{(n)}, Y_i^{(n)})\right\}$. Each $X_i^{(k)} \in \mathcal{X}_i \subseteq \mathcal{X}$, where $\mathcal{X}_i$ denotes the local input space at user i and $\mathcal{X}$ denotes a global input space which satisfies $\mathcal{X} \subseteq \cup_{i=1}^N \mathcal{X}_i$. The samples $\left\{X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(n)}\right\}$ are i.i.d generated according to a distribution P_i. Furthermore, the k-th label $Y_i^{(k)}$ of user i is given by

$$Y_i^{(k)} = f_{\theta^*}\left(X_i^{(k)}\right) + \eta_i^{(k)}, \tag{1}$$

where f_{θ^*} is a function defined over the input space $\mathcal{X}$, θ^* denotes a fixed global "true parameter" belonging to the set Θ which is a compact subset of $\mathbb{R}^d$ and $\eta_i^{(k)}$ denotes independent additive noise at user i. The global true parameter θ^* is unknown to the users. We assume that each user knows its *local likelihood functions* of the labels $\{\theta \in \Theta, x \in \mathcal{X}_i : l_i(\cdot; \theta, x)\}$, where $l_i(\cdot; \theta, x)$ denotes the local likelihood function of the label given θ is the true parameter and x in the input. Hence, the samples in the dataset for each user are conditionally independent and identically distributed (i.i.d) according to the distribution $f_i(x, y; \theta) = \mathsf{P}_i(x) l_i(y; \theta, x)$ but the observations might be correlated across the users.

Define $\bar{\Theta}_i = \{\theta \in \Theta \setminus \{\theta^*\}, x \in \mathcal{X}_i : l_i(\cdot; \theta, x) = l_i(\cdot; \theta^*, x)\}$. Note that, if the local input space $\mathcal{X}_i$ is such that $\bar{\Theta}_i \neq \phi$, then θ^* is not uniquely learnable using the local observations of user i. In this work, we are interested in the case where $\bar{\Theta}_i \neq \phi$ for some user i, but the true hypothesis θ^* is *globally learnable*, i.e, $\cap_{i=1}^N \bar{\Theta}_i = \phi$. Hence, the users aim to collaboratively learn underlying parameter θ^*.

Assumption 1. *The true parameter θ^* is globally learnable and hence $\cap_{i=1}^N \bar{\Theta}_i = \phi$.*

Example 1 (Distributed Linear Regression)**.** Suppose $d \geq 2$, $\Theta = [0, 1]^{d+1}$ and $\mathcal{X} = \mathbb{R}^d$. For any $\theta \in \Theta$, $x \in \mathcal{X}$, define $f_\theta(x) = \theta_0 + \sum_{i=1}^d \theta_i x_i = \langle \theta, [1, x^T]^T \rangle$. Suppose the observation noise is Gaussian given by $\eta \sim N(0, \alpha^2)$. For some $0 < m < d$, let $\mathcal{X}_1 = \left\{ \begin{bmatrix} x_1 \\ 0 \end{bmatrix} \mid x_1 \in \mathbb{R}^m \right\}$ and let $\mathcal{X}_2 = \left\{ \begin{bmatrix} 0 \\ x_2 \end{bmatrix} \mid x_2 \in \mathbb{R}^{d-m} \right\}$. Consider a simple network of two users (denoted by A and B), such that user A can make observations corresponding to points in $\mathcal{X}_1$ and user B in $\mathcal{X}_2$. For any $\theta \in \Theta$, define $\theta_0 \in [0, 1]$, $\theta_{(1:m)} \in [0, 1]^m$ and $\theta_{(m+1,d)} \in [0, 1]^{d-m}$ such that $\theta = [\theta_0, \theta_{(1:m)}^T, \theta_{(m+1:d)}^T]^T$. Thus, working in the realizable setting with the true parameter being θ^*, user A can learn the set $\Theta_A = \left\{\theta \in \Theta \mid \theta_{(1:m)} = \theta^*_{(1:m)}\right\}$ and similarly user B can learn the set $\Theta_B = \left\{\theta \in \Theta \mid \theta_{(m+1,d)} = \theta^*_{(m+1,d)}\right\}$. The goal in the distributed learning framework is to design an information exchange rule which results in both users eventually learning the true parameter $\{\theta^*\} = \Theta_A \cap \Theta_B$.

We model the communication network between users via a directed graph with vertex set $[N]$. We define the neighborhood of user i, denoted by $\mathcal{N}(i)$, as the set of all users j who have an edge going to j to i. Furthermore, if user $j \in \mathcal{N}(i)$, it can send messages to user i. The social interaction of the users is characterized by a stochastic matrix W. The weight $W_{ij} \in [0, 1]$ is strictly positive if and only if $j \in \mathcal{N}(i)$ and $W_{ii} = 1 - \sum_{j=1}^N W_{ij}$. The weight W_{ij} denotes the confidence user i

has on the information it receives from user j. We make the following assumption that allows the information gathered at every user to be disseminated throughout the network.

Assumption 2. *The network is a strongly connected aperiodic graph. Hence, W is aperiodic and irreducible.*

We discretize the parameter space Θ with M representative points and denote the set of these points by Θ_M. For every user $i \in [N]$, let $\Theta_M^{(i)} = \text{argmin}_{\theta \in \Theta_M} \mathbb{E}_{\mathsf{P}_i} \left[D_{\text{KL}} \left(l_i(\cdot; \theta^*, X_i) || l_i(\cdot; \theta, X_i) \right) \right]$. Let $\Theta_M^* := \cap_{i=1}^N \Theta_M^{(i)}$. Since θ^* is the true parameter, we have $\Theta_M^* \neq \phi$ for any M points. At every instant k each user i maintains a private belief vector $\boldsymbol{\rho_i^{(k)}}$ and a public belief vector $\mathbf{b_i^{(k)}}$, which are probability distributions on Θ.

Assumption 3. *For all users $i \in [N]$ we assume*

- *The prior beliefs $\rho_i^{(0)}(\theta) > 0$ for all $\theta \in \Theta_M$.*
- *There exists an $\alpha, L > 0$ such that $\alpha < l_i(y; \theta, x) < L$, for all $y \in \mathcal{Y}$, $\theta \in \Theta$ and $x \in \mathcal{X}_i$.*

We say that an algorithm learns θ in a distribution manner if the following holds: for any discretized parameter space Θ_M, for any $\delta \in (0, 1)$ we have

$$\mathsf{P}(\exists\, i \in [N] \text{ s.t. } \hat{\theta}_i^{(n)} \neq \theta_M^*) \leq \delta,$$

where $\hat{\theta}_i^{(n)}$ denotes the estimate of user i using n samples. Our learning criteria is requires every user in the network to agree on a parameter that best the observations over the entire network.

3 Distributed Learning Algorithm

We employ the distributed hypothesis testing algorithm considered by Lalitha et al. (2018); Shahrampour et al. (2016); Nedić et al. (2015) to cooperatively learn the model over the network. Suppose each user i starts with an initial private belief vector $\boldsymbol{\rho_i^{(0)}}$. At each instant $k \in [n]$ the following events happen:

1. Each user i draws an i.i.d sample $X_i^{(k)} \sim \mathsf{P}_i$ and obtains a conditionally i.i.d sample label $Y_i^{(k)} \sim \mathsf{P}_i \left(X_i^{(k)} \right) l_i \left(\cdot; \theta^*, X_i^{(k)} \right)$.
2. Each user i performs a local Bayesian update on $\boldsymbol{\rho_i^{(k-1)}}$ to form $\mathbf{b_i^{(k)}}$ using the following rule. For each $\theta \in \Theta_M$,
$$b_i^{(k)}(\theta) = \frac{l_i \left(Y_i^{(k)}; \theta, X_i^{(k)} \right) \rho_i^{(k-1)}(\theta)}{\sum_{\psi \in \Theta_M} l_i \left(Y_i^{(k)}; \psi, X_i^{(k)} \right) \rho_i^{(k-1)}(\psi)}. \tag{2}$$
3. Each user i sends the message $\mathbf{b_i^{(k)}}$ to all users j for which $i \in \mathcal{N}(j)$. Similarly receives messages from its neighbors $\mathcal{N}(i)$.
4. Each user i updates its private belief of every $\theta \in \Theta_M$, by averaging the log beliefs it received from its neighbors. For each $\theta \in \Theta_M$,
$$\rho_i^{(k)}(\theta) = \frac{\exp \left(\sum_{j=1}^N W_{ij} \log b_j^{(k)}(\theta) \right)}{\sum_{\psi \in \Theta_M} \exp \left(\sum_{j=1}^N W_{ij} \log b_j^{(k)}(\psi) \right)}. \tag{3}$$
5. At $k = n$, each user declares an estimate $\hat{\theta}_i^{(n)} = \text{argmax}_{\theta \in \Theta_M} \rho_i^{(n)}(\theta)$.

Note that the private belief vector $\boldsymbol{\rho_i^{(k)}}$ remain locally with the users while their public belief vectors $\mathbf{b_i^{(k)}}$ are exchanged with the neighbors.

Theorem 1. *Given a finite set Θ_M with M parameters. Fix some $\theta^*_M \in \Theta^*_M$. Using the distributed learning algorithm described above, for any given confidence $\delta \in (0,1)$ we have*

$$\mathsf{P}(\exists\, i \in [N] \textit{ s.t. } \hat{\theta}_i^{(n)} \neq \theta^*_M) \leq \delta,$$

when the number of samples satisfies

$$n \geq \max\left\{ \frac{16C\log N}{K(\Theta_M)(1-\lambda_{max}(W))}, \frac{16C\log\frac{NM}{\delta}}{K(\Theta_M)^2}, \frac{2\log(2M)}{K(\Theta_M)} \right\}, \tag{4}$$

where we define

$$K(\Theta_M) := \min_{\theta,\psi\in\Theta_M:\theta\neq\psi} \sum_{j=1}^{N} v_j I_j(\theta,\psi),$$

and define

$$I_j(\theta,\psi) := \mathbb{E}_{\mathsf{P}_j}\left[D_{KL}\left(l_j(\cdot;\theta^*,X_j)||l_j(\cdot;\psi,X_j)\right) - D_{KL}\left(l_j(\cdot;\theta^*,X_j)||l_j(\cdot;\theta,X_j)\right)\right],$$

where $\mathbf{v} = [v_1, v_2, \ldots, v_N]$ the unique stationary distribution of W with strictly positive components, $\lambda_{max}(W) = \max_{1\leq i\leq N-1}\lambda_i(W)$, $\lambda_i(W)$ denotes eigenvalue of W counted with algebraic multiplicity and $\lambda_0(W) = 1$ and $C = \frac{L}{\alpha}$.

Remark 1. The lower bound on the number of samples grows logarithmically in the number of users in the network and number of parameters to be distinguished. The lower bound also inversely depends on $K(\theta_M)$ which dictates the smallest rate at which the users distinguishes between true parameter and the wrong parameters across the network.

Let $r_i(x,y)$ denote the risk function of user $i \in [N]$ associated with sample $(x,y) \in \mathcal{D}_i$. The expected risk at user i when θ is the underlying parameter is given by $R_i(\theta) = \mathbb{E}_{\mathsf{P}_i}\left[\int_{\mathcal{Y}} r_i(x,y) l_i\left(y;\theta,x\right) dy\right]$. Now, using the above theorem we obtain the following bounds on the average expected risk over the network as a corollary.

Corollary 1. *Suppose the parameter space Θ consists of countably many parameters. For any $r > 0$, define $\mathcal{B}_r(\theta) = \left\{\psi \in \Theta : \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\mathsf{P}_i}[D_{KL}(l_i\left(\cdot;\theta,x\right)||l_i\left(\cdot;\psi,x\right))] \leq r\right\}$. Consider $\Theta_M \subset \Theta$ of cardinality M, such that $\Theta \subset \cup_{\theta\in\Theta_M} B_r(\theta)$. Fix some $\theta^*_M \in \Theta^*_M$. If $|r_i(x,y)| \leq B$ for all $x \in \mathcal{X}_i$, $y \in \mathcal{Y}$, then using the above algorithm with probability at least $1-\delta$ for the number of samples given by* (4) *we have*

$$\begin{aligned}
\frac{1}{N}\sum_{i=1}^{N} |R_i(\theta^*) - R_i(\theta^*_M)| &\leq \frac{B}{N}\sum_{i=1}^{N} \mathbb{E}_{\mathsf{P}_i}\left[\int_{\mathcal{Y}} |l_i\left(y;\theta^*,x\right) - l_i\left(y;\theta^*_M,x\right)|\, dy\right] \\
&\overset{(a)}{\leq} \frac{B}{2N}\sum_{i=1}^{N} \mathbb{E}_{\mathsf{P}_i}\left[\sqrt{D_{KL}\left(l_i\left(y;\theta^*,x\right)||l_i\left(y;\theta^*_M,x\right)\right)}\right] \\
&\overset{(b)}{\leq} \frac{B}{2}\sqrt{\frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\mathsf{P}_i}\left[D_{KL}\left(l_i\left(y;\theta^*,x\right)||l_i\left(y;\theta^*_M,x\right)\right)\right]} \\
&\overset{(c)}{\leq} \frac{B\sqrt{r}}{2},
\end{aligned}$$

where (a) follows from Pinsker's inequality, (b) from Jensen's inequality and (c) follows from Theorem 1.

A similar corollary can be obtained for a continuous parameter set Θ by using the Hellinger distance to construct an r-covering and by using the relation between Hellinger distance and total variational distance.

Example 1 Revisited (Distributed Bayesian Linear Regression)**.** Now we embed the users in an aperiodic network with edge weights given by $W = \begin{bmatrix} 0.9 & 0.1 \\ 0.6 & 0.4 \end{bmatrix}$. Fix $d = 2$, $m = 1$ and

$\theta^* = [-0.3,\ 0.5,\ 0.8]^T$. In other words, $\Theta = \mathbb{R}^3$ and $\mathcal{X} = \mathbb{R}^2$. Suppose the observation noise is distributed as $\eta \sim \mathcal{N}(0, \alpha^2)$ where $\alpha = 0.8$. User A makes observation corresponding to x_1, i.e., in $\mathbb{R}$ and user B makes observations corresponding to x_2, i.e., in $\mathbb{R}$. We assume each user starts with a Gaussian prior over θ with zero mean $[0, 0, 0]^T$ and covariance matrix given by $\begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$. Furthermore, x_1 is sampled from Unif$[-1, 1]$ which can be accessed by user A only and x_2 is sampled from Unif$[-1.5, 1.5]$ which can be accessed by user B only. However, the test set consists of observations corresponding to $\mathcal{X}$.

Suppose there exists a central user which has access to a training set which consists of observations corresponding to $\mathcal{X}$. This central user obtains a posterior on θ using the training set samples and uses the posterior distribution to make a Bayesian prediction on the samples in the test set. Then, Figure 1(a) shows Mean Squared Error (MSE) of the prediction over the test set for the central user. For the distributed setup, there are two cases:

1. Training without cooperation among users: In this case we assume that each user uses only the local observations to train, hence obtains the posterior distribution on θ using the local training samples only. Figure 1(a) shows MSE of the Bayesian prediction over the test set for both users for this case. Comparing Figures 1(a) and (b) we can see that MSE of both users is higher than that of central user implying the performance of users has degraded due to insufficient local information to learn θ^*.
2. Training using the proposed distributed learning rule: In this case we assume that each user uses the local observations and the posterior distribution on θ obtained from its neighboring users to train. The exchange and merge of posterior distribution among neighboring users is dictated by our distributed learning rule. Figure 1(c) shows MSE of the prediction over the test set for both users for this case. Comparing Figures 1(a) and (c) we can see that MSE of both users matches that of central user implying using the proposed learning rule to train the users were able to learn θ^*.

Remark 2. Note that likelihood functions considered in Example 1 violate the bounded likelihood functions assumption since the likelihood functions in it are Gaussian. Furthermore, the parameters belong to a continuous parameter set Θ. This example demonstrates that our analytical assumptions on the likelihood functions and the parameter set are not necessary for convergence of our distributed learning rule.

3.1 Application to Training DNNs

Each iteration of the algorithm described above involves a local Bayesian update (2), followed by a consensus step (3). Exactly computing the normalizing constants in these update rules is computationally intractable for most practical problems. We propose the following modifications to make the general scheme proposed above suitable for learning DNN models.

- For the local Bayesian update rule, we can employ Variational Inference (VI) (Gal, 2016, Chapter 3) techniques to obtain an approximate posterior at each user. This involves solving the following objective function over a parametrized class of distributions ($q_\varphi(\cdot)$).

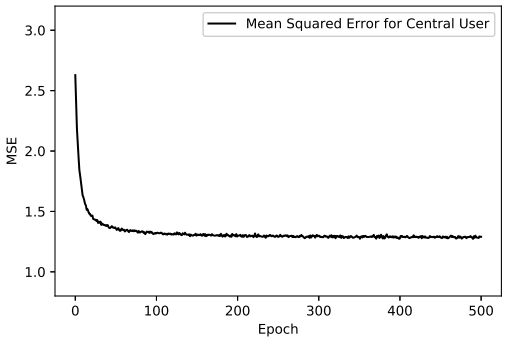
$$\mathcal{L}_{VI}(\theta) \coloneqq \int_\Theta q_\varphi(\theta) \log l_i\left(y; \theta, x\right) d\theta + D_{\text{KL}}\left(q_\varphi(\theta) || \rho_i^{(k)}(\theta)\right) \tag{5}$$

- Furthermore, Since the private belief (i.e., $\rho_i^{(k)}(\cdot)$ only appears in the KL Divergence term in (5), using an unnormalized form of belief vector $\boldsymbol{\rho}_i$ does not alter the optimization problem. More specifically, for any $\kappa > 0$, we have

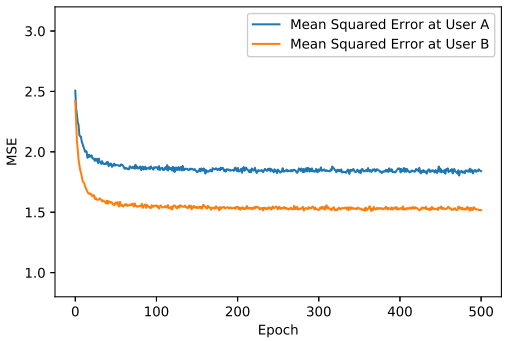
$$\begin{aligned} D_{\text{KL}}\left(q_\varphi(\theta) || \kappa\rho_i^{(k)}(\theta)\right) &= D_{\text{KL}}\left(q_\varphi(\theta) || \rho_i^{(k)}(\theta)\right) + \int q_\varphi(\theta) \log \kappa \, d\theta \\ &= D_{\text{KL}}\left(q_\varphi(\theta) || \rho_i^{(k)}(\theta)\right) + \log \kappa \end{aligned}$$

  Thus we can perform the consensus update with the unnormalized beliefs.

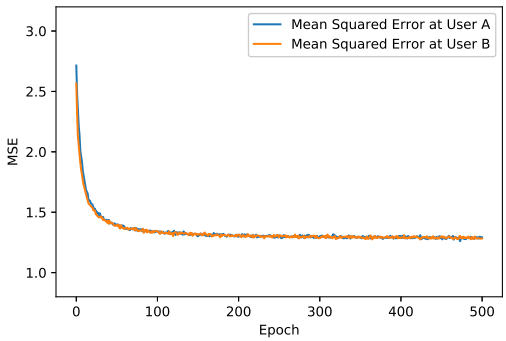
- Furthermore, instead of performing updates after every observed sample, a batch of observations can be used for obtaining the approximate posterior update using VI techniques.



(a)



(b)



(c)

Figure 1: Figure shows the MSE of Bayesian prediction on the test set samples over time for three cases: (a) a central user trained with samples across the network, (b) user A and B trained only using the local observations with no cooperation among the users (c) user A and B trained using the proposed distributed learning rule which combines local observations with the posterior over the parameter θ received from the neighboring users.

4 Discussion and Future Work

In this paper, we considered the problem of decentralized learning over a network of users with no central server. Moreover, our framework allows the individual users only sample points from small subspaces of the input space. The users take Bayesian-like approach via the introduction of a belief on the parameter space. We considered a learning scheme in which users iterate and aggregate the beliefs of their one-hop neighbors and collaboratively estimate the global optimal parameter. We obtained high probability bounds on the network wide worst case probability of error, and also discussed suitable approximations for applying this algorithm for learning DNN models.

An important area of future work is to conduct empirical studies to evaluate the performance of the proposed algorithm in learning DNN models.

References

Gal, Y. (2016). Uncertainty in deep learning. *University of Cambridge*.

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. (2016). Federated Learning: Strategies for Improving Communication Efficiency. *arXiv preprint arXiv:1610.05492*.

Lalitha, A., Javidi, T., and Sarwate, A. D. (2018). Social Learning and Distributed Hypothesis Testing. *IEEE Transactions on Information Theory*, 64(9):6161–6179.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Artificial Intelligence and Statistics*, pages 1273–1282.

Nedić, A., Olshevsky, A., and Uribe, C. A. (2015). Non-asymptotic Convergence Rates for Cooperative Learning over Time-varying Directed Graphs. In *2015 American Control Conference (ACC)*, pages 5884–5889.

Shahrampour, S., Rakhlin, A., and Jadbabaie, A. (2016). Distributed Detection: Finite-Time Analysis and Impact of Network Topology. *IEEE Transactions on Automatic Control*, 61(11):3256–3268.

5 Appendix

5.1 Proof of Theorem 1

The proof of Theorem 1 is based the proof provided by Lalitha et al. (2018). For the ease of exposition, let $\rho_i^{(0)}(\theta) = \frac{1}{M}$ for all $\theta \in \Theta_M$. Fix some $\theta_M^* \in \Theta_M^*$. We begin with the following recursion for each user i and any $\theta \in \Theta_M \setminus \theta_M^*$,

$$\frac{1}{n}\log\frac{\rho_i^{(n)}(\theta_M^*)}{\rho_i^{(n)}(\theta)} = \frac{1}{n}\sum_{j=1}^{N}\sum_{k=1}^{n} W_{ij}^k z_j^{(n-k+1)}(\theta_M^*, \theta),$$

where

$$z_j^{(k)}(\theta_M^*, \theta) = \log\frac{l_j\left(Y_j^{(k)}; \theta_M^*, X_i^{(k)}\right)}{l_j\left(Y_j^{(k)}; \theta, X_i^{(k)}\right)}.$$

From the above recursion we have

$$\begin{aligned}
\frac{1}{n}\log\frac{\rho_i^{(n)}(\theta_M^*)}{\rho_i^{(n)}(\theta)} &= \frac{1}{n}\sum_{j=1}^{N}\left(\sum_{k=1}^{n}\left(W_{ij}^k - v_j\right) z_j^{(n-k+1)}(\theta_M^*, \theta)\right) + \frac{1}{n}\sum_{j=1}^{N} v_j\left(\sum_{k=1}^{n} z_j^{(n-k+1)}(\theta_M^*, \theta)\right) \\
&\geq -\frac{1}{n}\sum_{j=1}^{N}\sum_{k=1}^{n}\left|W_{ij}^k - v_j\right|\left|z_j^{(k)}(\theta_M^*, \theta)\right| + \frac{1}{n}\sum_{j=1}^{N} v_j\left(\sum_{k=1}^{n} z_j^{(k)}(\theta_M^*, \theta)\right) \\
&\overset{(a)}{\geq} -\frac{4C\log N}{n(1-\lambda_{\max}(W))} + \frac{1}{n}\sum_{j=1}^{N} v_j\left(\sum_{k=1}^{n} z_j^{(k)}(\theta_M^*, \theta)\right),
\end{aligned}$$

where (a) follows from Lemma 1 and the boundedness assumption of log-likelihood ratios. Now fix $n \geq \frac{8C\log N}{\epsilon(1-\lambda_{\max}(W))}$, since $\rho_i^{(n)}(\theta_M^*) \leq 1$ we have

$$-\frac{1}{n}\log\rho_i^{(n)}(\theta) \geq -\frac{\epsilon}{2} + \frac{1}{n}\sum_{j=1}^{N} v_j\left(\sum_{k=1}^{n} z_j^{(k)}(\theta_M^*, \theta)\right).$$

Furthermore, we have

$$\begin{aligned}
&\mathsf{P}\left(-\frac{1}{n}\log\rho_i^{(n)}(\theta) \leq \sum_{j=1}^{N} v_j I_j(\theta_M^*, \theta) - \epsilon\right) \\
&\leq \mathsf{P}\left(\frac{1}{n}\sum_{j=1}^{N} v_j\left(\sum_{k=1}^{n} z_j^{(k)}(\theta_M^*, \theta)\right) \leq \sum_{j=1}^{N} v_j I_j(\theta_M^*, \theta) - \frac{\epsilon}{2}\right),
\end{aligned}$$

where

$$\begin{aligned}
I_j(\theta_M^*, \theta) &= \mathbb{E}[z_j(\theta_M^*, \theta)] \\
&= \mathbb{E}_{\mathsf{P}_j}\left[D_{\mathrm{KL}}\left(l_j(\cdot;\theta^*, X_j)||l_j(\cdot;\theta, X_j)\right)\right] - \mathbb{E}_{\mathsf{P}_j}\left[D_{\mathrm{KL}}\left(l_j(\cdot;\theta^*, X_j)||l_j(\cdot;\theta_M^*, X_j)\right)\right].
\end{aligned}$$

Now for any $j \in [N]$ note that

$$\sum_{j=1}^{N} v_j \sum_{k=1}^{n} z_j^{(k)}(\theta_M^*, \theta) = \sum_{k=1}^{n}\left(\sum_{j=1}^{N} v_j z_j^{(k)}(\theta_M^*, \theta) - \sum_{j=1}^{N} v_j \mathbb{E}[z_j^{(k)}(\theta_M^*, \theta)]\right) + n\sum_{j=1}^{N} v_j I_j(\theta_M^*, \theta).$$

For any $\theta \in \Theta_M \setminus \theta_M^*$, applying McDiarmid's inequality for any $\epsilon > 0$ and for all $n \geq 1$ we have

$$\mathsf{P}\left(\sum_{k=1}^{n}\left(\sum_{j=1}^{N} v_j z_j^{(k)}(\theta_M^*, \theta) - \sum_{j=1}^{N} v_j \mathbb{E}[z_j^{(k)}(\theta_M^*, \theta)]\right) \leq -\frac{\epsilon n}{2}\right) \leq e^{-\frac{\epsilon^2 n}{2C}}.$$

This implies for all $\theta \in \Theta_M \setminus \theta_M^*$,

$$\mathsf{P}\left(\frac{1}{n}\sum_{j=1}^{N} v_j \left(\sum_{k=1}^{n} z_j^{(k)}(\theta_M^*, \theta)\right) \leq \sum_{j=1}^{N} v_j I_j(\theta_M^*, \theta) - \frac{\epsilon}{2}\right) \leq e^{-\frac{\epsilon^2 n}{2C}}$$

Hence, for all $\theta \in \Theta_M \setminus \theta_M^*$, for $n \geq \frac{8C \log N}{\epsilon(1-\lambda_{\max}(W))}$ we have

$$\mathsf{P}\left(\frac{-1}{n}\log \rho_i^{(n)}(\theta) \leq \sum_{j=1}^{N} v_j I_j(\theta_M^*, \theta) - \epsilon\right) \leq e^{-\frac{\epsilon^2 n}{4C}},$$

which implies

$$\mathsf{P}\left(\rho_i^{(n)}(\theta) \geq e^{-n(\sum_{j=1}^{N} v_j I_j(\theta_M^*, \theta) - \epsilon)}\right) \leq e^{-\frac{\epsilon^2 n}{4C}}.$$

Using this we obtain a bound on the worst case error over all θ and across the entire network as follows

$$\mathsf{P}\left(\max_{i\in[N]} \max_{\theta \in \Theta_M \setminus \theta_M^*} \rho_i^{(n)}(\theta) \geq e^{-n(K(\Theta_M)-\epsilon)}\right) \leq NMe^{-\frac{\epsilon^2 n}{4C}},$$

where $K(\Theta_M) := \min_{\theta,\psi \in \Theta_M : \theta \neq \psi} \sum_{j=1}^{N} v_j I_j(\theta, \psi)$. Choose $\epsilon = \frac{K(\Theta_M)}{2}$. Therefore, for a given confidence $\delta \in (0, 1)$ we have

$$\mathsf{P}_{\text{error}} \leq \mathsf{P}\left(\max_{i\in[N]} \max_{\theta \in \Theta_M \setminus \theta_M^*} \rho_i^{(n)}(\theta) \geq \frac{1}{2M}\right) \leq \delta,$$

when the number of samples satisfies

$$n \geq \max\left\{\frac{16C \log N}{K(\Theta_M)(1-\lambda_{\max}(W))}, \frac{16C \log \frac{NM}{\delta}}{K(\Theta_M)^2}, \frac{2\log(2M)}{K(\Theta_M)}\right\}.$$

Lemma 1 (Shahrampour et al. (2016))**.** *For a strongly connected aperiodic network, the Markov chain with transition probabilities given by W is irreducible and aperiodic, and the unique stationary distribution $\mathbf{v} = [v_1, v_2, \ldots, v_N]$ has strictly positive components and satisfies*

$$v_i = \sum_{j=1}^{n} v_j W_{ji}.$$

Furthermore, for any $i \in [N]$ the weight matrix satisfies

$$\sum_{k=1}^{n}\sum_{j=1}^{N} \left|W_{ij}^k - v_j\right| \leq \frac{4 \log N}{1 - \lambda_{max}(W)},$$

where $\lambda_{max}(W) = \max_{1\leq i\leq N-1} \lambda_i(W)$, where $\lambda_i(W)$ denotes eigenvalue of W counted with algebraic multiplicity and $\lambda_0(W) = 1$.