Federated Algorithm With An Exponential Weighted Average Approach: Fed-Exp

Sai Anuroop Kesanapalli* B. N. Bharath†

April 15, 2021

1 Federated Learning

1.1 Introduction

Federated Learning (FL) is a distributed machine learning architecture where edge-devices learn shared predictive model collaboratively. FL actually started as a project at Google in 2017. In this architecture, an edge-device downloads the current model, improves it by learning from its local data, and then summarizes the changes as a small focused update. Only this update to the model is sent to the federating server in the cloud, where it is processed with updates from other edge-devices to improve the shared model. All the data remains on the edge-device, hence privacy is preserved.
We start with listing some applications of FL in the next section.

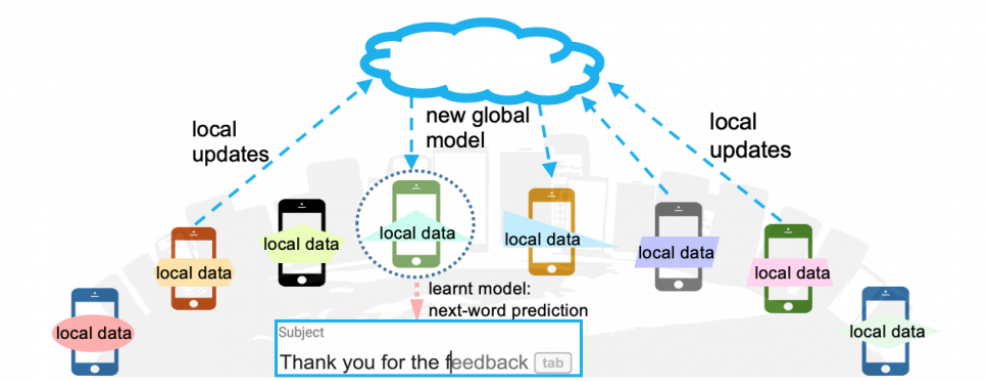


Figure 1: FL Architecture
https://blog.ml.cmu.edu/2019/11/12/federated-learning-challenges-methods-and-future-directions/

1.2 Applications of Federated Learning

Google currently uses FL in their Android GBoard to predict the next query a user makes based on the words he types. In many European Union countries, patient privacy laws are stringent. It is legally burdensome to send patient data such as MRI-scans, X-ray films or blood-samples to the cloud for diagnostic services based on machine learning algorithms. In such scenarios, FL can be employed for private learning among various hospitals, with the data staying with the hospitals itself. Another promising application of FL is self-driving vehicles in an autonomous vehicle network. Let us say that in near future we have self-driving cars on roads, and the cluster of these cars within a geographical perimeter can collaboratively learn the traffic conditions.
We next highlight some of the challenges in FL.

*Department of Computer Science and Engineering, Indian Institute of Technology Dharwad
†Department of Electrical Engineering, Indian Institute of Technology Dharwad

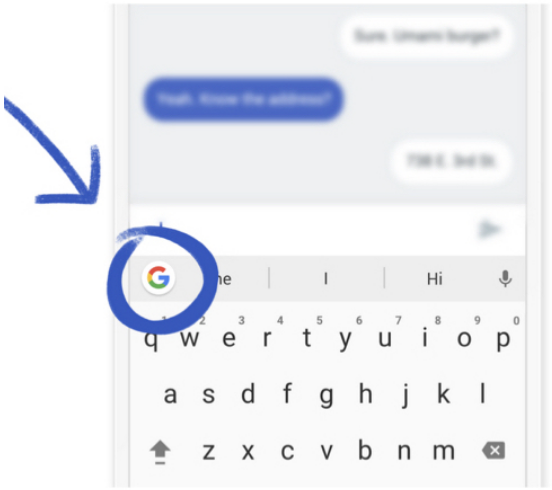


Figure 2: GBoard: Query prediction using FL
https://blog.google/products/search/gboard-now-on-android/

1.3 Challenges in Federated Learning

There are five main challenges in FL which are described below:

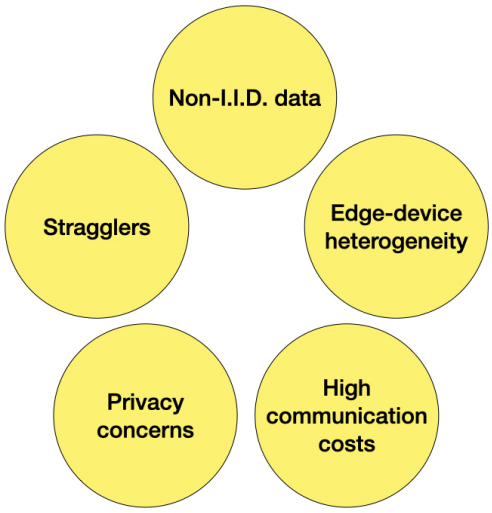


Figure 3: Big-five challenges in FL

1.3.1 Stragglers:

First challenge is about stragglers in the edge-devices. Let us take the example of smartphones learning a task collaboratively. Some phones may not be available at the time the learning is happening. In such cases, how does the model that is learnt perform due to the unavailability of these phones?

1.3.2 Non-IID data:

Next is the assumption that the data available at edge-devices in non-IID. The assumption that data is IID makes analysis simpler, however, doing so may incur heavy errors in learning. In principle, that data available at different edge-devices may follow different distributions altogether.

1.3.3 Device heterogeneity:

We then have device heterogeneity. As an example, smartphones vary in their memory, processor capacity and network bandwidth available to them. They are all not the same. How do we account for this while creating our model?

1.3.4 Communication costs:

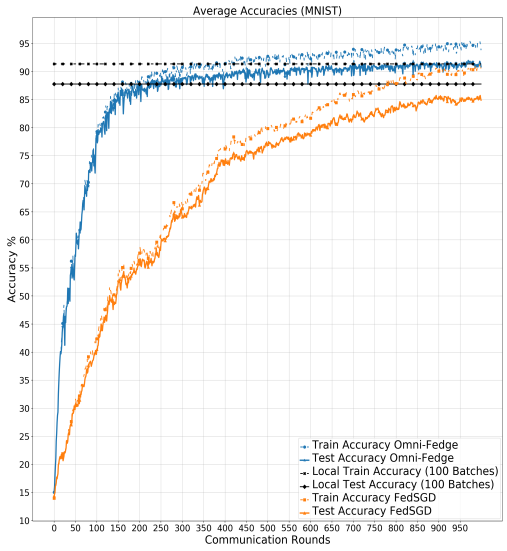
Since federated learning is a distributed machine learning architecture, there will be lots of communication taking place between various entities in the network. So communication costs will be high. How do we optimise the communication?

1.3.5 Privacy concerns:

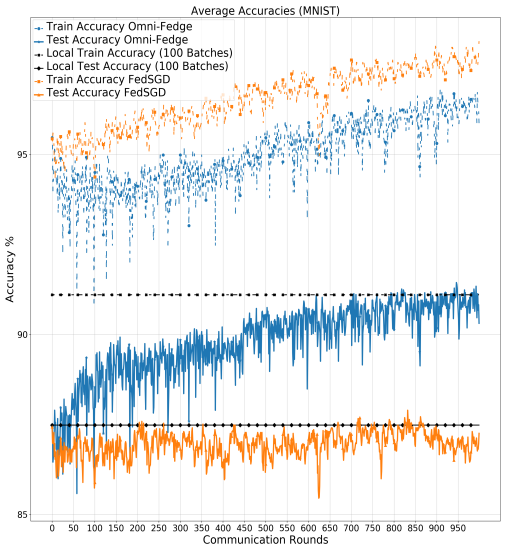
Finally, we have the privacy concerns. Though federated learning is privacy preserving by default, how do we ensure that the updates cannot be re-traced to identify the users? How do we encrypt them?
In the following section, we present our previous work (BTP-I) on FL.

2 Our previous work on Federated Learning

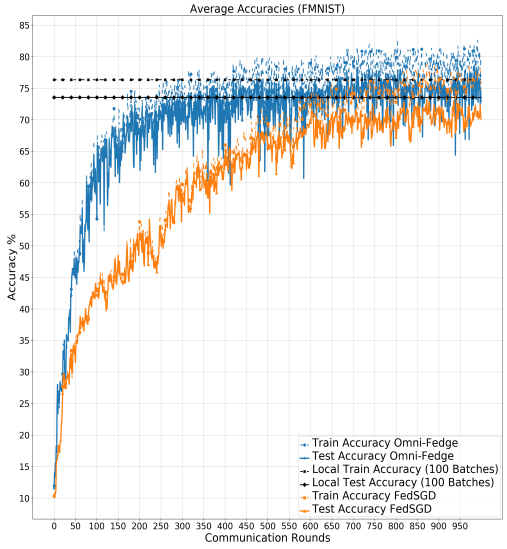
In our previous work, we considered the problem of FL under non-IID data setting. We provided an improved estimate of the empirical loss at each node by using a weighted average of losses across nodes with a penalty term. These uneven weights to different nodes were assigned by taking a novel Bayesian approach to the problem where the problem of learning for each device/node was cast as maximizing the likelihood of a joint distribution. This joint distribution was for losses of nodes obtained by using data across devices for a given neural network of a node. We then provided a PAC learning guarantee on the objective function which revealed that the true average risk was no more than the proposed objective and the error term. We leveraged this guarantee to propose an algorithm called *Omni-Fedge*. Using MNIST and Fashion MNIST data-sets, we showed that the performance of the proposed algorithm is significantly better than existing algorithms. We then submitted a paper titled *"Federated Algorithm With Bayesian Approach: Omni-Fedge"* to IEEE ICASSP 2021, which has been accepted for presentation at the conference. We next introduce the concept of online learning.



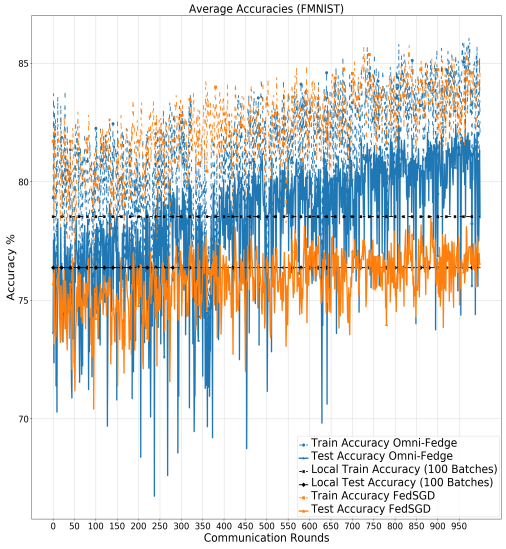
(a) MNIST non-i.i.d



(b) MNIST i.i.d



(c) FMNIST non-i.i.d



(d) FMNIST i.i.d

Figure 4: Plots of Average Accuracies vs Communication Rounds for Omni-Fedge and FedSGD

3 Online Learning

3.1 Background to Online Learning

Online learning algorithms are suitable for modern applications since they provide an efficient solution for large-scale problems. These algorithms process one sample at a time with an update per iteration that is often computationally cheap and easy to implement. On-line algorithms do not require any distributional assumption: their analysis assumes an adversarial scenario. Note that this is in stark contrast to our previous work, as there is no notion of generalisation in the Online Learning scenario.
Consider the following example on weather prediction which illustrates online learning. Let us call the each of

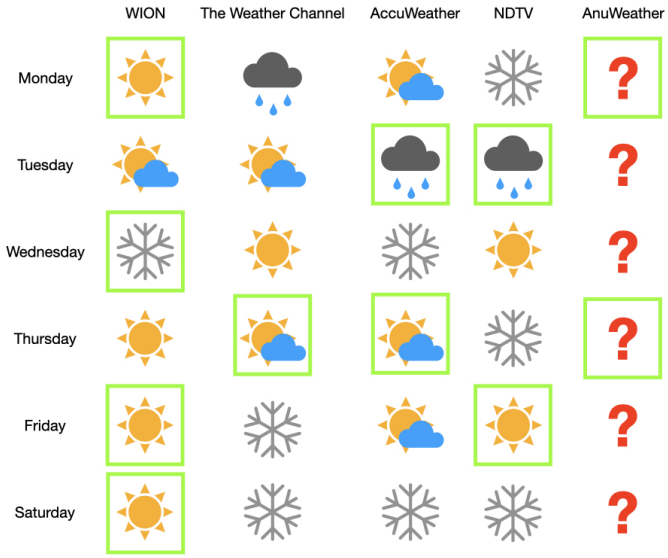


Figure 5: Weather prediction example

the weather forecasting agencies shown in the figure *experts*. For simplicty, assume that they provide us with their prediction of the weather once every day. Let the prediction of i^{th} expert for t^{th} day be $\hat{y}_{i,t}$. Let us say our algorithm predicts the weather using the information obtained from the experts, call that $\hat{y}_t$. After the prediction is done for the all the days of a given week, the true weather conditions for those days is now known. Denote the true condition of t^{th} day as y_t. We would now like to know in the *hindsight* how well our algorithm had predicted the weather conditions when compared to the experts. In doing so, we compare ourselves with the best expert, i.e., the one who had predicted better than the rest.
In the following subsection, we define the notion of regret, as we will be using it in our current work.

3.2 Regret

More formally, the objective in this setting is to minimize the regret R_T, which compares the cumulative loss of the algorithm to that of the best expert in *hindsight* after T rounds of prediction:

$$R_T = \sum_{t=1}^{T} \mathcal{L}(\hat{y}_t, y_t) - \min_{i=1}^{N} \sum_{t=1}^{T} \mathcal{L}(\hat{y}_{i,t}, y_t)$$

In the following section, we present our work (BTP-II).

4 Fed-Exp

We now consider the problem of online learning in a federated setup. Here, the task for each edge-device/node is to learn the neural network in a federated fashion which performs well on the local data and as well captures the global trend by exploiting the statisitcal similarity of data available across different nodes in the network.

4.1 Motivation

Our objective is to refine the neural network of a node by considering it as a linear combination of neural networks of the other nodes, weighed by the statistical similarity of the data available at these nodes. We thus

require a simple method to weigh the neural networks learnt by various nodes in the network. To this cause, we find that exponential weighted average is one such simple method which does the job. It is in particular suitable for online learning scenario where the neural network needs to adapt itself quickly to the incoming data. We have observed that though FL in online scenario is studied, most of these studies provide equal weight to the nodes in the network, rather than considering a weighted average approach. We note here that Fed-Exp is 'lighter' communication and computation-wise when compared to *Omni-Fedge*, our previous work, which shall be shortly evident.

4.2 Problem Setting

We now formally introduce the problem setting and various definitions required in our work.

- There are N edge-devices/nodes and one Federating Server (FS).
- Let $\hat{\theta}_{i,t}$ denote the neural network of i^{th} node at time t and $\mathcal{L}_{i,t}(\hat{\theta}_{j,t}, z_{i,t})$ denote the loss of node i at time t using the neural network of node j at time t, where $z_{i,t}$ denotes the data (or a batch of data points) seen by node i at time t.
- Let $\omega_{ij,t}$ denote the weight given by node i to node j based upon the loss incurred by node i on using the neural network of node j.

4.3 Algorithm

We now present the algorithm for *Fed-Exp*.
for $t \in \{1, \ldots, T\}$ **do**

- Update $\omega_{ij,t+1} \leftarrow \omega_{ij,t} e^{-\eta \mathcal{L}_{i,t}(\hat{\theta}_{j,t}, z_{i,t})}$, $\forall\ j \in \{1, \ldots, N\}$, where $\eta > 0$ is a hyper-parameter and $\omega_{ij,1} = 1\ \forall\ i, j \in \{1, \ldots, N\}$
- Compute $\Gamma_{i,t+1} = arg\min_{\Gamma} \mathcal{L}_{i,t+1}(\Gamma, z_{i,t+1})$ and broadcast to all nodes
- **if** $\left| \mathbb{E}_{\boldsymbol{p}_{i,t}}[(\hat{\theta}_{j,t} - \Gamma_{j,t})^T \nabla \mathcal{L}_{i,t}(\Gamma_{j,t}, z_{i,t})] \right| \leq \frac{c}{\sqrt{T}}$
  - Update $\hat{\theta}_{i,t+1} \leftarrow \frac{\sum_{j \in \mathcal{S}_{t+1}} \omega_{ij,t+1} \Gamma_{j,t+1}}{\sum_{j \in \mathcal{S}_{t+1}} \omega_{ij,t+1}}$ and broadcast to all nodes, where $\mathcal{S}_{t+1} \subseteq \{1, \ldots, N\}$ is randomly chosen and is of cardinality K
- **else**
  - Update $\hat{\theta}_{i,t+1} \leftarrow \Gamma_{j,t+1}$ and broadcast to all nodes, where $\mathcal{S}_{t+1} \subseteq \{1, \ldots, N\}$ is randomly chosen and is of cardinality K

An explanation of the intuition behind the proposed algorithm is provided next.

4.4 Explanation

In the first step of our algorithm, i^{th} node i updates the weight of j^{th} node by looking at the loss incurred by i^{th} node on using neural network of j^{th} node. Intuitively, we need to give less weight to that node whose neural network performs badly on i^{th} node. The exponential decay function just achieves that. Next, we find the best neural network for i^{th} node by looking at the data available to it in (or until) the t^{th} iteration. This step ensures that the neural network is customised to the stream of local data. Finally, we refine the neural network by updating it as the linear combination of neural networks obtained by various nodes in the previous step, weighed according to the weights obtained in the first step. Doing so will help us find the global trend. We note here that we have a weird scheme of retaining the refined neural network, which otherwise reverts back to the best local neural network obtained in the previous step. A basic explanation to this is that this step adds an $\mathcal{O}(T)$ term to the regret bound, which is not desirable as it would mean that the number of mistakes the algorithm makes is of the order of number of iterations - implying that there is no effective learning happening. Ideally, we would want the algorithm to make lesser mistakes as the number of iterations progresses. So, we make a test based on the inner product of the difference of the refined neural network and the best local neural network with the gradient of the loss function evaluated using the best local neural network. This would become more clear when we discuss proof of the theorem on regret bound.
We note here that the algorithm involves two broadcasts, one each after the second and the third steps. This is one broadcast less than that in our previous algorithm *Omni-Fedge*. Also, the proposed algorithm is computationally 'lighter' as the computation of weights here does not involve any objective minimisation and relatively

fewer gradient-computation steps. Finally, the random choice of nodes in the last step is to take into account the straggling nature of nodes and more importantly, to reduce the communication overhead of the algorithm - note that each broadcast amounts to an $\mathcal{O}(K^2)$ communication cost where K is the number active nodes in the current iteration of the algorithm. If the number of nodes N is large, communication becomes really expensive. In the following subsection, we provide theoretical guarantees for the proposed algorithm.

4.5 Theoretical Guarantees

We gauge the performance of the proposed algorithm using standard regret-based analysis of online learning algorithms, albeit customising it to our problem setting which is federated in nature. We begin with defining some more terms and assumptions, which under normal circumstances, are valid.

- Let $\omega_{ij,t+1} = \omega_{ij,t} e^{-\eta \mathcal{L}_{i,t}(\hat{\theta}_{j,t}, z_{i,t})} = e^{-\eta L_{j,t}}$.
- Define $\Phi_t = \log \sum_{j \in \mathcal{S}_t} \omega_{ij,t}$. We call this the *potential function.*
- Let $\boldsymbol{p}_{i,t}$ denote the probability distribution over $j = \{1, \ldots, N\}$ such that $p_{ij,t} = \dfrac{\omega_{ij,t}}{\sum_{j \in \mathcal{S}_t} \omega_{ij,t}}$.
- Assume that $\mathcal{S}_1 = \{1, \ldots, N\}$ and hence $\Phi_1 = \log N$.
- Assume that the loss function $\mathcal{L}_{i,t}$ is convex in its first argument and is $\mathcal{C}^1$.
- Assume that the gradient of the loss function is Lipschitz continuous.

We define regret for the proposed algorithm as follows:

$$\mathcal{R}_{i,T}^{Fed-Exp} = \mathbb{E}\Big[\sum_{t=1}^{T} \mathcal{L}_{i,t}(\hat{\theta}_{i,t}, z_{i,t})\Big] - \mathbb{E}[\min_{j \in \{1,\ldots,N\}} (L_{j,T})]$$

We present the main result of our work, which is a theorem on regret bound of the proposed algorithm.

Theorem 1. *Regret Bound of Fed-Exp*

$$\mathbb{E}\Big[\sum_{t=1}^{T} \mathcal{L}_{i,t}(\hat{\theta}_{i,t}, z_{i,t})\Big] - \mathbb{E}[\min_{j \in \{1,\ldots,N\}} (L_{j,T})] <$$

$$\frac{1}{\eta} \sum_{t=1}^{T} \log\Big(\frac{(K-1)^2 K}{N} e^{-\eta(L_{min,t} - L_{max,t})} + 1\Big) + \frac{\eta T}{8} + cK\sqrt{T} + \frac{\log N}{\eta} + \left(1 - \frac{K}{N}\right) L_T$$

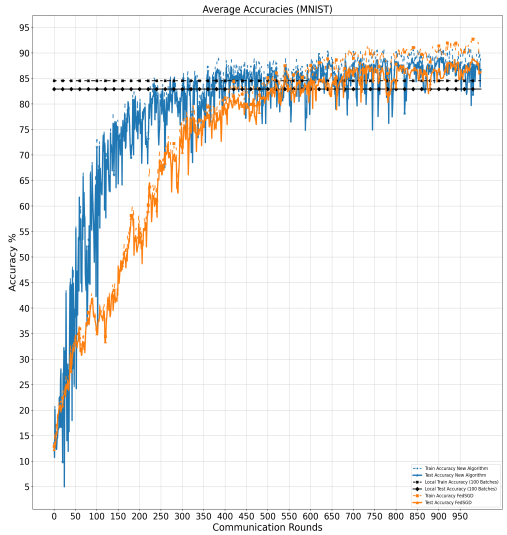
See Appendix for proof. We now present the experimental results using the proposed algorithm.

4.6 Experimental Results

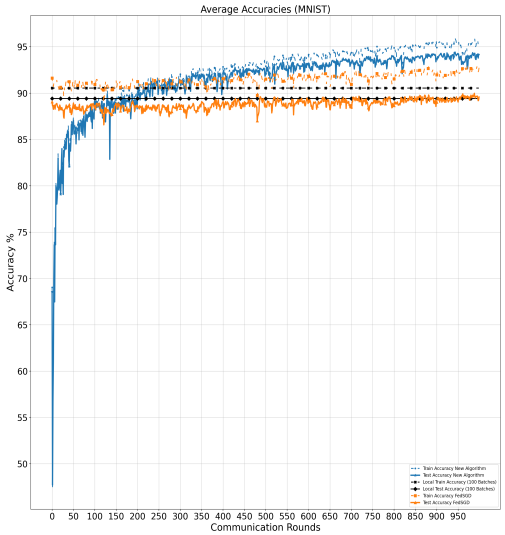
The experiments are carried out using MNIST handwritten data set and FMNIST fashion data set, with 5 nodes and a central FS. We have assumed that $K = N(= 5)$ and the experiments are done using the basic version of the proposed algorithm wherein the conditional step always updates the neural network by taking a linear combination of other neural networks. Both MNIST and FMNIST have $60,000$ training examples and a test set of $10,000$ examples each. Each example is a 28×28 grayscale image, associated with labels from 10 classes. The table below shows the split between training and test data. The non-i.i.d case is emulated using a non-uniform sampling of data from the MNIST and FMNIST data sets. The data samples assigned to each node are further divided into 400 batches. The proposed algorithm is compared with (i) local training, i.e., training using local data only, and (ii) FedSGD. In FedSGD, the gradient is averaged, and one neural network is used across all the devices. Fig. 7 shows the performance of the proposed algorithm compared with the above mentioned algorithms for both training and test data. The table in Fig. 6 shows the split between training and test data.

Type of Data	Training samples per node	Testing samples per node	Number of batches	Number of training samples per iteration	Number of testing samples per iteration
MNIST i.i.d.	2488	3112	400	~6.22	~7.78
MNIST n.i.i.d.	799	1000	400	~2	~2.5
FMNIST i.i.d.	3111	3889	400	~7.77	~9.72
FMNIST n.i.i.d.	971	1178	400	~2.42	~2.95

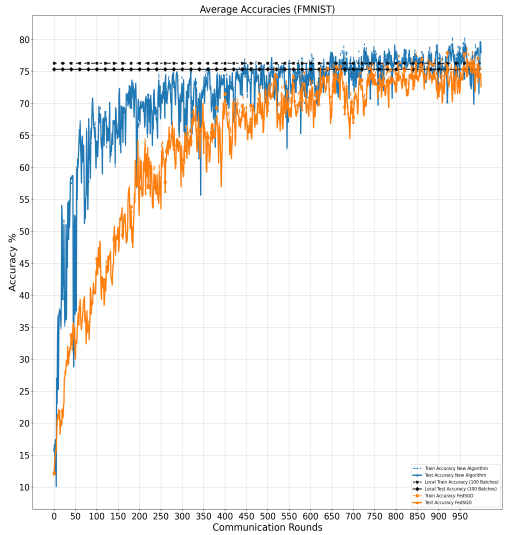
Figure 6: Train-test data split



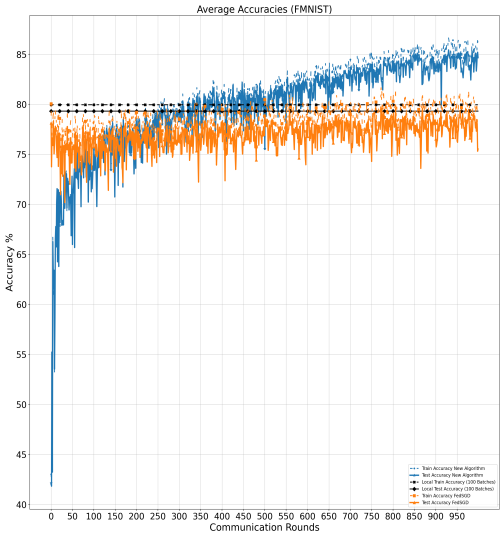
(a) MNIST non-i.i.d



(b) MNIST i.i.d



(c) FMNIST non-i.i.d



(d) FMNIST i.i.d

Figure 7: Plots of Average Accuracies vs Communication Rounds for Fed-Exp and FedSGD

5 Conclusion

We proposed a federated algorithm in an online learning scenario and provided a sound theoretical framework on its regret bound. We tested the proposed algorithm using MNIST and FMNIST data-sets and observed that the algorithm performs considerably better than FedSGD and local training in both the cases. We propose to further better the regret bound and study the performance of the proposed algorithm on other data-sets. Also, we envisage a large scale implementation of both of the algorithms that we have proposed.

6 Appendix

Note: Proof outlined here needs to be refined further.
By taking $\eta = \mathcal{O}(\frac{1}{\sqrt{T}})$, asymptotic analysis of the regret bound in the theorem suggests that the regret goes down as $\mathcal{O}(T\sqrt{T}\log\frac{K^3}{N}) + \mathcal{O}(\sqrt{T}) + \mathcal{O}(\sqrt{T}\log N) + \left(1 - \frac{K}{N}\right)L_T$ and its average goes down as $\mathcal{O}(\sqrt{T}\log\frac{K^3}{N}) + \mathcal{O}(\frac{1}{\sqrt{T}}) + \mathcal{O}(\frac{\log N}{\sqrt{T}}) + \left(1 - \frac{K}{N}\right)\frac{L_T}{T}$, which is not that desirable as we would expect the regret to go down as $\mathcal{O}(\sqrt{T})$.

Proof. We make use of the potential function described earlier to derive upper and lower bounds for this function, and combine them to obtain our result. This potential function method is a general proof technique which is used quintessentially in analysis of regret bounds.

We begin with finding the upper bound. Consider,

$$\mathbb{E}[\Phi_{t+1} - \Phi_t] = \mathbb{E}\left[\log\left(\sum_{j\in\mathcal{S}_{t+1}} \omega_{ij,t+1}\right) - \log\left(\sum_{j\in\mathcal{S}_t} \omega_{ij,t}\right)\right]$$

$$= \mathbb{E}\left[\log\left(\frac{\sum_{j\in\mathcal{S}_{t+1}} \omega_{ij,t+1}}{\sum_{j\in\mathcal{S}_t} \omega_{ij,t}}\right)\right]$$

Multiplying and dividing by $\sum_{j\in\mathcal{S}_t} \omega_{ij,t+1}$, as $\omega_{ij,t+1} = \omega_{ij,t}e^{-\eta\mathcal{L}_{i,t}(\hat{\theta}_{j,t},z_{i,t})}$, and as $p_{ij,t} = \dfrac{\omega_{ij,t}}{\sum_{j\in\mathcal{S}_t} \omega_{ij,t}}$,

$$= \mathbb{E}\left[\log\left(\frac{\sum_{j\in\mathcal{S}_{t+1}} \omega_{ij,t+1}}{\sum_{j\in\mathcal{S}_t} \omega_{ij,t+1}}\right)\left(\sum_{j\in\mathcal{S}_t} p_{ij,t}e^{-\eta\mathcal{L}_{i,t}(\hat{\theta}_{j,t},z_{i,t})}\right)\right]$$

$$= \mathbb{E}\left[\log\left(\frac{\sum_{j\in\mathcal{S}_{t+1}} \omega_{ij,t+1}}{\sum_{j\in\mathcal{S}_t} \omega_{ij,t+1}}\right)\left(\mathbb{E}_{\boldsymbol{p}_{i,t}}[e^{\eta X}]\right)\right]$$

where $X = -\mathcal{L}_{i,t}(\hat{\theta}_{j,t}, z_{i,t})$

$$= \mathbb{E}\left[\log\left(\frac{\sum_{j\in\mathcal{S}_{t+1}} \omega_{ij,t+1}}{\sum_{j\in\mathcal{S}_t} \omega_{ij,t+1}}\right) + \log\left(\mathbb{E}_{\boldsymbol{p}_{i,t}}[e^{\eta X}]\right)\right]$$

We now split each of $\mathcal{S}_{t+1}$ and $\mathcal{S}_t$ into two parts such that one contains elements exlcusive only to either of them and the other contains elements common to both

$$= \mathbb{E}\left[\log\left(\frac{\sum_{j\in\mathcal{S}_{t+1}\setminus\mathcal{S}_t} \omega_{ij,t+1} + \sum_{j\in\mathcal{S}_{t+1}\cap\mathcal{S}_t} \omega_{ij,t+1}}{\sum_{j\in\mathcal{S}_t\setminus\mathcal{S}_{t+1}} \omega_{ij,t+1} + \sum_{j\in\mathcal{S}_t\cap\mathcal{S}_{t+1}} \omega_{ij,t+1}}\right) + \log\left(\mathbb{E}_{\boldsymbol{p}_{i,t}}[e^{\eta X}]\right)\right]$$

$$= \mathbb{E}\left[\log\frac{\left(\frac{\sum_{j\in\mathcal{S}_{t+1}\setminus\mathcal{S}_t} \omega_{ij,t+1}}{\sum_{j\in\mathcal{S}_{t+1}\cap\mathcal{S}_t} \omega_{ij,t+1}}\right) + 1}{\left(\frac{\sum_{j\in\mathcal{S}_t\setminus\mathcal{S}_{t+1}} \omega_{ij,t+1}}{\sum_{j\in\mathcal{S}_t\cap\mathcal{S}_{t+1}} \omega_{ij,t+1}}\right) + 1} + \log\left(\mathbb{E}_{\boldsymbol{p}_{i,t}}[e^{\eta X}]\right)\right]$$

as $\omega_{ij,t+1} = e^{-\eta L_{j,t}}$,

$$= \mathbb{E}\left[\log\frac{\left(\frac{\sum_{j\in\mathcal{S}_{t+1}\setminus\mathcal{S}_t} e^{-\eta L_{j,t}}}{\sum_{j\in\mathcal{S}_{t+1}\cap\mathcal{S}_t} e^{-\eta L_{j,t}}}\right) + 1}{\left(\frac{\sum_{j\in\mathcal{S}_t\setminus\mathcal{S}_{t+1}} e^{-\eta L_{j,t}}}{\sum_{j\in\mathcal{S}_t\cap\mathcal{S}_{t+1}} e^{-\eta L_{j,t}}}\right) + 1} + \log\left(\mathbb{E}_{\boldsymbol{p}_{i,t}}[e^{\eta X}]\right)\right]$$

$$\leq \mathbb{E}\left[\log\frac{\left(\frac{|\mathcal{S}_{t+1}\setminus\mathcal{S}_t|e^{-\eta L_{min,t}}}{|\mathcal{S}_{t+1}\cap\mathcal{S}_t|e^{-\eta L_{max,t}}}\right) + 1}{\left(\frac{|\mathcal{S}_t\setminus\mathcal{S}_{t+1}|e^{-\eta L_{max,t}}}{|\mathcal{S}_t\cap\mathcal{S}_{t+1}|e^{-\eta L_{min,t}}}\right) + 1} + \log\left(\mathbb{E}_{\boldsymbol{p}_{i,t}}[e^{\eta X}]\right)\right]$$

$$= \mathbb{E}\left[\log\frac{\left(\frac{|\mathcal{S}_{t+1}\setminus\mathcal{S}_t|}{|\mathcal{S}_{t+1}\cap\mathcal{S}_t|}e^{-\eta(L_{min,t}-L_{max,t})}\right) + 1}{\left(\frac{|\mathcal{S}_t\setminus\mathcal{S}_{t+1}|}{|\mathcal{S}_t\cap\mathcal{S}_{t+1}|}e^{\eta(L_{min,t}-L_{max,t})}\right) + 1} + \log\left(\mathbb{E}_{\boldsymbol{p}_{i,t}}[e^{\eta X}]\right)\right]$$

By linearity of expectation,

$$= \mathbb{E}\left[\log \frac{\left(\frac{|\mathcal{S}_{t+1} \setminus \mathcal{S}_t|}{|\mathcal{S}_{t+1} \cap \mathcal{S}_t|} e^{-\eta(L_{min,t} - L_{max,t})}\right) + 1}{\left(\frac{|\mathcal{S}_t \setminus \mathcal{S}_{t+1}|}{|\mathcal{S}_t \cap \mathcal{S}_{t+1}|} e^{\eta(L_{min,t} - L_{max,t})}\right) + 1}\right] + \mathbb{E}\left[\log\left(\mathbb{E}_{\boldsymbol{p}_{i,t}}[e^{\eta X}]\right)\right]$$

As log is concave and by applying Jensen's inequality to the argument of log function,

$$\leq \log \frac{\left(\mathbb{E}\left[\frac{|\mathcal{S}_{t+1} \setminus \mathcal{S}_t|}{|\mathcal{S}_{t+1} \cap \mathcal{S}_t|}\right] e^{-\eta(L_{min,t} - L_{max,t})}\right) + 1}{\left(\mathbb{E}\left[\frac{|\mathcal{S}_t \setminus \mathcal{S}_{t+1}|}{|\mathcal{S}_t \cap \mathcal{S}_{t+1}|}\right] e^{\eta(L_{min,t} - L_{max,t})}\right) + 1} + \mathbb{E}\left[\log\left(\mathbb{E}_{\boldsymbol{p}_{i,t}}[e^{\eta X}]\right)\right]$$

Since $|\mathcal{S}_{t+1}| = |\mathcal{S}_t| = K$, as $\mathbb{P}\Big[|\mathcal{S}_t \cap \mathcal{S}_{t+1}| = m\Big] = \frac{^{N-m}C_{K-m}}{^{N}C_K}$, we have

$$\mathbb{E}[\Phi_{t+1} - \Phi_t] \leq \log \frac{\left(\sum_{m=1}^{K-1}\left(\frac{K-m}{m}\right)\frac{^{N-m}C_{K-m}}{^{N}C_K} e^{-\eta(L_{min,t} - L_{max,t})}\right) + 1}{\left(\sum_{m=1}^{K-1}\left(\frac{K-m}{m}\right)\frac{^{N-m}C_{K-m}}{^{N}C_K} e^{\eta(L_{min,t} - L_{max,t})}\right) + 1} + \mathbb{E}\left[\log\left(\mathbb{E}_{\boldsymbol{p}_{i,t}}[e^{\eta X}]\right)\right] \tag{1}$$

Let us further upper bound the following term

$$\log \frac{\left(\sum_{m=1}^{K-1}\left(\frac{K-m}{m}\right)\frac{^{N-m}C_{K-m}}{^{N}C_K} e^{-\eta(L_{min,t} - L_{max,t})}\right) + 1}{\left(\sum_{m=1}^{K-1}\left(\frac{K-m}{m}\right)\frac{^{N-m}C_{K-m}}{^{N}C_K} e^{\eta(L_{min,t} - L_{max,t})}\right) + 1}$$

Observe that

$$\begin{aligned} \frac{^{N-m}C_{K-m}}{^{N}C_K} &= \prod_{i=1}^{m} \frac{K-i+1}{N-i+1} \\ &\text{As } 0 \leq K \leq N \\ &\leq \left(\frac{K}{N}\right)^m \ \forall\ m \in \{1, \dots K-1\} \end{aligned} \quad ,$$

So,

$$\begin{aligned} \sum_{m=1}^{K-1}\left(\frac{K-m}{m}\right)\frac{^{N-m}C_{K-m}}{^{N}C_K} &\leq \sum_{m=1}^{K-1}\left(\frac{K-m}{m}\right)\left(\frac{K}{N}\right)^m \\ &\leq \frac{(K-1)^2 K}{N} \end{aligned} \tag{2}$$

Also observe that,

$$\begin{aligned} &\text{As } 0 \leq K \leq N\ , \\ &\frac{^{N-m}C_{K-m}}{^{N}C_K} = \prod_{i=1}^{m} \frac{K-i+1}{N-i+1} > 0\ \forall\ m \in \{1, \dots K-1\} \end{aligned}$$

So,

$$\sum_{m=1}^{K-1}\left(\frac{K-m}{m}\right)\frac{^{N-m}C_{K-m}}{^{N}C_K} > \sum_{m=1}^{K-1}\left(\frac{K-m}{m}\right).0 = 0 \tag{3}$$

Using (2) and (3) in $\log \dfrac{\left(\sum_{m=1}^{K-1}\left(\frac{K-m}{m}\right)\frac{^{N-m}C_{K-m}}{^{N}C_{K}}e^{-\eta(L_{min,t}-L_{max,t})}\right)+1}{\left(\sum_{m=1}^{K-1}\left(\frac{K-m}{m}\right)\frac{^{N-m}C_{K-m}}{^{N}C_{K}}e^{\eta(L_{min,t}-L_{max,t})}\right)+1}$, we get

$$\log \frac{\left(\sum_{m=1}^{K-1}\left(\frac{K-m}{m}\right)\frac{^{N-m}C_{K-m}}{^{N}C_{K}}e^{-\eta(L_{min,t}-L_{max,t})}\right)+1}{\left(\sum_{m=1}^{K-1}\left(\frac{K-m}{m}\right)\frac{^{N-m}C_{K-m}}{^{N}C_{K}}e^{\eta(L_{min,t}-L_{max,t})}\right)+1} < \log\left(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\right) \tag{4}$$

Using (4) in (1) we get,

$$\begin{aligned}
\mathbb{E}[\Phi_{t+1}-\Phi_t] &< \log\left(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\right)+\mathbb{E}\left[\log\left(\mathbb{E}_{\boldsymbol{p}_{i,t}}[e^{\eta X}]\right)\right] \qquad (5)\\
&= \log\left(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\right)+\mathbb{E}\Big[\log\left(\mathbb{E}_{\boldsymbol{p}_{i,t}}[e^{\eta(X-\mathbb{E}_{\boldsymbol{p}_{i,t}}[X])+\eta\mathbb{E}_{\boldsymbol{p}_{i,t}}[X]}]\right)\Big]\\
&\text{By Hoeffding's Lemma,}\\
&\le \log\left(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\right)+\frac{\eta^2}{8}+\eta\mathbb{E}\Big[\mathbb{E}_{\boldsymbol{p}_{i,t}}[X]\Big]\\
&= \log\left(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\right)+\frac{\eta^2}{8}-\eta\mathbb{E}\Big[\mathbb{E}_{\boldsymbol{p}_{i,t}}[\mathcal{L}_{i,t}(\hat{\theta}_{j,t},z_{i,t})]\Big]\\
&\text{Since,}\\
&\hat{\theta}_{j,t}=\begin{cases}\frac{\sum_{k\in\mathcal{S}_t}\omega_{jk,t}\Gamma_{k,t}}{\sum_{k\in\mathcal{S}_t}\omega_{jk,t}}\ (=\hat{\theta}^*_{j,t}), & \text{if } \left|\mathbb{E}_{\boldsymbol{p}_{i,t}}[(\hat{\theta}_{j,t}-\Gamma_{j,t})^T\nabla\mathcal{L}_{i,t}(\Gamma_{j,t},z_{i,t})]\right|\le\frac{c}{\sqrt{T}}\\ \Gamma_{j,t}, & \text{otherwise}\end{cases}\\
&\text{We define an indicator function } \mathcal{I}_j \text{ such that}\\
&\mathcal{I}_j=\begin{cases}0, & \text{if } \left|\mathbb{E}_{\boldsymbol{p}_{i,t}}[(\hat{\theta}_{j,t}-\Gamma_{j,t})^T\nabla\mathcal{L}_{i,t}(\Gamma_{j,t},z_{i,t})]\right|\le\frac{c}{\sqrt{T}}\\ 1, & \text{otherwise}\end{cases}\\
&= \log\left(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\right)+\frac{\eta^2}{8}-\eta\mathbb{E}\Big[\mathbb{E}_{\boldsymbol{p}_{i,t}}[\mathcal{I}_j\mathcal{L}_{i,t}(\Gamma_{j,t},z_{i,t})+(1-\mathcal{I}_j)(\mathcal{L}_{i,t}(\hat{\theta}^*_{j,t},z_{i,t})]\Big]\\
&= \log\left(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\right)+\frac{\eta^2}{8}-\eta\mathbb{E}\Big[\mathbb{E}_{\boldsymbol{p}_{i,t}}[\mathcal{I}_j\mathcal{L}_{i,t}(\Gamma_{j,t},z_{i,t})\\
&+(1-\mathcal{I}_j)(\mathcal{L}_{i,t}(\Gamma_{j,t}+\hat{\theta}^*_{j,t}-\Gamma_{j,t},z_{i,t})]\Big]\\
&\text{By assuming } \mathcal{L}_{i,t} \text{ is } \mathcal{C}^1 \text{ ,}\\
&\le \log\left(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\right)+\frac{\eta^2}{8}-\eta\mathbb{E}\Big[\mathbb{E}_{\boldsymbol{p}_{i,t}}[\mathcal{I}_j\mathcal{L}_{i,t}(\Gamma_{j,t},z_{i,t})\\
&+(1-\mathcal{I}_j)(\mathcal{L}_{i,t}(\Gamma_{j,t},z_{i,t})+(\hat{\theta}^*_{j,t}-\Gamma_{j,t})^T\nabla\mathcal{L}_{i,t}(\Gamma_{j,t},z_{i,t}))]\Big]\\
&= \log\left(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\right)+\frac{\eta^2}{8}-\eta\mathbb{E}\Big[\mathbb{E}_{\boldsymbol{p}_{i,t}}[\mathcal{L}_{i,t}(\Gamma_{j,t},z_{i,t})\\
&+(1-\mathcal{I}_j)(\hat{\theta}^*_{j,t}-\Gamma_{j,t})^T\nabla\mathcal{L}_{i,t}(\Gamma_{j,t},z_{i,t})]\Big]\\
&\text{By linearity of expectation,}\\
&= \log\left(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\right)+\frac{\eta^2}{8}-\eta\mathbb{E}\Big[\mathbb{E}_{\boldsymbol{p}_{i,t}}[\mathcal{L}_{i,t}(\Gamma_{j,t},z_{i,t})]\\
&+\mathbb{E}_{\boldsymbol{p}_{i,t}}[(1-\mathcal{I}_j)(\hat{\theta}^*_{j,t}-\Gamma_{j,t})^T\nabla\mathcal{L}_{i,t}(\Gamma_{j,t},z_{i,t})]\Big]\\
&\le \log\left(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\right)+\frac{\eta^2}{8}-\eta\mathbb{E}\Big[\mathbb{E}_{\boldsymbol{p}_{i,t}}[\mathcal{L}_{i,t}(\Gamma_{j,t},z_{i,t})]\\
&-\mathbb{E}_{\boldsymbol{p}_{i,t}}[(1-\mathcal{I}_j)(\hat{\theta}^*_{j,t}-\Gamma_{j,t})^T\nabla\mathcal{L}_{i,t}(\Gamma_{j,t},z_{i,t})]\Big]
\end{aligned}$$

By linearity of expectation,

$$= \log\Big(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\Big)+\frac{\eta^2}{8}-\eta\mathbb{E}\Big[\mathbb{E}_{\boldsymbol{p}_{i,t}}[\mathcal{L}_{i,t}(\Gamma_{j,t},z_{i,t})]\Big]$$
$$+\eta\mathbb{E}\Big[\mathbb{E}_{\boldsymbol{p}_{i,t}}[(1-\mathcal{I}_j)(\hat{\theta}^*_{j,t}-\Gamma_{j,t})^T\nabla\mathcal{L}_{i,t}(\Gamma_{j,t},z_{i,t})]\Big]$$
$$\leq \log\Big(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\Big)+\frac{\eta^2}{8}-\eta\mathbb{E}\Big[\mathbb{E}_{\boldsymbol{p}_{i,t}}[\mathcal{L}_{i,t}(\Gamma_{j,t},z_{i,t})]\Big]+\eta\mathbb{E}\Big[\mathbb{E}_{\boldsymbol{p}_{i,t}}\Big[(1-\mathcal{I}_j)\frac{c}{\sqrt{T}}\Big]\Big]$$
$$= \log\Big(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\Big)+\frac{\eta^2}{8}-\eta\mathbb{E}\Big[\mathbb{E}_{\boldsymbol{p}_{i,t}}[\mathcal{L}_{i,t}(\Gamma_{j,t},z_{i,t})]\Big]+\eta\frac{c}{\sqrt{T}}\mathbb{E}\Big[\mathbb{E}_{\boldsymbol{p}_{i,t}}[(1-\mathcal{I}_j)]\Big]$$

By convexity of loss function in the first argument,

$$\leq \log\Big(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\Big)+\frac{\eta^2}{8}-\eta\mathbb{E}\Big[\mathcal{L}_{i,t}(\mathbb{E}_{\boldsymbol{p}_{i,t}}[\Gamma_{j,t}],z_{i,t})\Big]+\eta\frac{c}{\sqrt{T}}\mathbb{E}\Big[\mathbb{E}_{\boldsymbol{p}_{i,t}}[(1-\mathcal{I}_j)]\Big]$$

As $\hat{\theta}_{i,t}=\mathbb{E}_{\boldsymbol{p}_{i,t}}[\Gamma_{j,t}]$,

$$= \log\Big(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\Big)+\frac{\eta^2}{8}-\eta\mathbb{E}\Big[\mathcal{L}_{i,t}(\hat{\theta}_{i,t},z_{i,t})\Big]+\eta\frac{c}{\sqrt{T}}\mathbb{E}\Big[\mathbb{E}_{\boldsymbol{p}_{i,t}}[(1-\mathcal{I}_j)]\Big]$$
$$\leq \log\Big(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\Big)+\frac{\eta^2}{8}-\eta\mathbb{E}\Big[\mathcal{L}_{i,t}(\hat{\theta}_{i,t},z_{i,t})\Big]+\eta\frac{cK}{\sqrt{T}}$$

By linearity of expectation,

$$\mathbb{E}[\Phi_{t+1}]-\mathbb{E}[\Phi_t]<\log\Big(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\Big)+\frac{\eta^2}{8}-\eta\mathbb{E}\Big[\mathcal{L}_{i,t}(\hat{\theta}_{i,t},z_{i,t})\Big]+\eta\frac{cK}{\sqrt{T}}$$

By summing over all $t\in\{1,\ldots,T\}$, we finally get the upper bound as follows

$$\mathbb{E}[\Phi_{T+1}]-\mathbb{E}[\Phi_1]<\sum_{t=1}^{T}\log\Big(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})}+1\Big)+\frac{\eta^2T}{8}-\eta\mathbb{E}\Big[\sum_{t=1}^{T}\mathcal{L}_{i,t}(\hat{\theta}_{i,t},z_{i,t})\Big]+\eta cK\sqrt{T}\quad(6)$$

We now find the lower bound. Again consider

$$
\begin{aligned}
&\text{As } \Phi_1 = \log N \ , \\
\mathbb{E}[\Phi_{T+1}] - \mathbb{E}[\Phi_1] &= \mathbb{E}[\log\big(\sum_{j\in\mathcal{S}_{T+1}} \omega_{ij,T+1}\big)] - \log N \\
&\text{As } \omega_{ij,T+1} = e^{-\eta L_{j,T}} \ , \\
&= \mathbb{E}[\log\big(\sum_{j\in\mathcal{S}_{T+1}} e^{-\eta L_{j,T}}\big)] - \log N \\
&\geq \mathbb{E}[\log\big(\max_{j\in\mathcal{S}_{T+1}} e^{-\eta L_{j,T}}\big)] - \log N \\
&= -\eta\mathbb{E}[\min_{j\in\mathcal{S}_{T+1}}(L_{j,T})] - \log N \\
&\text{Adding and subtracting } \min_{j\in\{1,\ldots,N\}}(L_{j,T}) \ , \\
&= -\eta\mathbb{E}[\min_{j\in\mathcal{S}_{T+1}}(L_{j,T}) - \min_{j\in\{1,\ldots,N\}}(L_{j,T})] - \eta\mathbb{E}[\min_{j\in\{1,\ldots,N\}}(L_{j,T})] - \log N \\
&= -\eta\Big[\mathbb{P}[arg\min_{j\in\{1,\ldots,N\}}(L_{j,T}) \in \mathcal{S}_{T+1}][\min_{j\in\{1,\ldots,N\}}(L_{j,T}) - \min_{j\in\{1,\ldots,N\}}(L_{j,T})] \\
&+ \mathbb{P}(arg\min_{j\in\{1,\ldots,N\}}(L_{j,T}) \notin \mathcal{S}_{T+1}][\min_{j\in\mathcal{S}_{T+1}}(L_{j,T}) - \min_{j\in\{1,\ldots,N\}}(L_{j,T})]\Big] \\
&- \eta\mathbb{E}[\min_{j\in\{1,\ldots,N\}}(L_{j,T})] - \log N \\
&= -\eta\left(1 - \frac{{}^{N-1}C_{K-1}}{{}^{N}C_{K}}\right)\Big[\min_{j\in\mathcal{S}_{T+1}}(L_{j,T}) - \min_{j\in\{1,\ldots,N\}}(L_{j,T})\Big] - \eta\mathbb{E}[\min_{j\in\{1,\ldots,N\}}(L_{j,T})] - \log N \\
&\text{As } \frac{{}^{N-1}C_{K-1}}{{}^{N}C_{K}} = \frac{K}{N} \ , \\
&= -\eta\left(1 - \frac{K}{N}\right)\Big[\min_{j\in\mathcal{S}_{T+1}}(L_{j,T}) - \min_{j\in\{1,\ldots,N\}}(L_{j,T})\Big] - \eta\mathbb{E}[\min_{j\in\{1,\ldots,N\}}(L_{j,T})] - \log N \\
&\geq -\eta\left(1 - \frac{K}{N}\right)L_T - \eta\mathbb{E}[\min_{j\in\{1,\ldots,N\}}(L_{j,T})] - \log N \\
&\text{where } \Big[\min_{j\in\mathcal{S}_{T+1}}(L_{j,T}) - \min_{j\in\{1,\ldots,N\}}(L_{j,T})\Big] \leq L_T
\end{aligned}
$$

So we have the lower bound as,

$$\mathbb{E}[\Phi_{T+1}] - \mathbb{E}[\Phi_T] \geq -\eta\left(1 - \frac{K}{N}\right)\Big[\min_{j\in\mathcal{S}_{T+1}}(L_{j,T}) - \min_{j\in\{1,\ldots,N\}}(L_{j,T})\Big] - \eta\mathbb{E}[\min_{j\in\{1,\ldots,N\}}(L_{j,T})] - \log N \tag{7}$$

Combining both the bounds in 6 and 7 yields,

$$
\begin{aligned}
-\eta\left(1 - \frac{K}{N}\right)L_T - \eta\mathbb{E}[\min_{j\in\{1,\ldots,N\}}(L_{j,T})] - \log N < \sum_{t=1}^{T}\log\Big(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})} + 1\Big) + \frac{\eta^2 T}{8} \\
-\eta\mathbb{E}\Big[\sum_{t=1}^{T}\mathcal{L}_{i,t}(\hat{\theta}_{i,t}, z_{i,t})\Big] + \eta cK\sqrt{T}
\end{aligned}
$$

On re-arranging terms, we get

$$
\begin{aligned}
\mathbb{E}\Big[\sum_{t=1}^{T}\mathcal{L}_{i,t}(\hat{\theta}_{i,t}, z_{i,t})\Big] - \mathbb{E}[\min_{j\in\{1,\ldots,N\}}(L_{j,T})] < \frac{1}{\eta}\sum_{t=1}^{T}\log\Big(\frac{(K-1)^2K}{N}e^{-\eta(L_{min,t}-L_{max,t})} + 1\Big) + \frac{\eta T}{8} \\
+cK\sqrt{T} + \frac{\log N}{\eta} + \left(1 - \frac{K}{N}\right)L_T
\end{aligned} \tag{8}
$$

□