On the Impact of Data Collection Strategies in Streaming Federated Learning with Markovian Data

Tan-Khiem HUYNH[1] Malcolm EGAN[1] Giovanni NEGLIA[2] Jean-Marie GORCE[1]

[1]INRIA, Université de Lyon, CITI, INSA Lyon, 69100 Villeurbanne, France

[2]INRIA, Université Côte d'Azur, 06560 Valbonne, France

Résumé – L'apprentissage fédéré (FL) est un cadre essentiel pour l'apprentissage collaboratif entre dispositifs périphériques, où les données sont souvent collectées de manière continue et présentent une dépendance statistique. Dans ce travail, nous comparons les performances des algorithmes FL dans deux configurations : l'une où les données des clients sont échantillonnées consécutivement à partir d'une chaîne de Markov, et l'autre où les échantillons entre les rounds de communication sont indépendants, bien que les échantillons au sein du même round restent dépendants. Ce dernier scénario est particulièrement pertinent dans les systèmes de FL où les contraintes de collection de données empêchent les clients de mettre continuellement à jour leurs mémoires. Nos expériences montrent que garantir l'indépendance entre les tours de communication améliore significativement la convergence, en particulier pour les processus de Markov à mélange lent.

Abstract – Federated Learning (FL) is a key framework for collaborative learning across edge devices, where data is often collected in a streaming manner and exhibits statistical dependence. In this work, we study the impact of such dependence by comparing the performance of FL algorithms under two settings: one where client data is sampled consecutively from a Markov chain and another where samples across communication rounds are independent, while remaining dependent within the same round. The latter setting is particularly relevant in FL systems where data collection constraints prevent clients from continuously updating their memory caches. Our experiments show that ensuring independence between rounds significantly improves convergence, particularly for slow-mixing Markov processes.

1 Introduction

Clients in edge networks are increasingly diverse, with a wide range of data collection and learning capabilities. In particular, many clients have access to streaming data rather than fixed historical data sets. This is often the case in health monitoring, robotics, and environmental tracking, where new data is collected over time.

At the same time, data from a single client is often not sufficient for learning in applications such as public health and resource management, which require aggregation of data across multiple individuals. Since transmitting sensitive or large datasets is costly, centralized training is impractical. An alternative approach, known as Federated Learning (FL), enables communication-efficient, collaborative learning by training models locally and aggregating them centrally [5].

Motivated by these factors, FL with streaming data has recently been investigated in the case of independently drawn samples [8]. However, data in health and environmental monitoring applications is often generated by non-linear physical and biological systems and modeled as non-stationary Markovian data streams [1]. Moreover, memory constraints mean that clients can only store a small number of samples. As a consequence, there is a need for FL algorithms which perform well with Markovian data, where samples are not drawn independently and access to data is limited by memory constraints.

While first-order optimization methods with Markovian sampling have been extensively studied in the centralized setting [2], recent work on Markovian data in FL has primarily focused on Federated Reinforcement Learning (FRL) with linear function approximation. A key challenge is demonstrating that collaboration is beneficial in this setting; namely, that the per-client sample complexity of FRL algorithms decreases inversely with the number of clients. While this linear speed-up is well-understood for i.i.d. data, the impact of collaboration remains an open question in the presence of statistical dependence, as is the case with Markovian data streams. The work in [6] demonstrated the benefits of cooperation under a strong homogeneity assumption on the client environment. Heterogeneous settings have been considered in [3] with a linear speed-up established in [10] subject to heterogeneity assumptions, which were then relaxed in [7] by incorporating control variates.

In [4], we investigated the performance of FL algorithms with Markovian data streams in a broader learning context involving non-convex objectives, which commonly arise in classification and regression tasks in systems with non-linear dynamics [12]. Our results showed that collaboration in FL remains beneficial in this setting. However, the analysis in [4] does not explicitly account for the difference between two cases: one where client data is drawn sequentially from a Markov process, and one where samples in different communication rounds are independent. We highlight that in the latter case, samples within the same communication rounds are still drawn consecutively from a Markov chain. This setting is particularly relevant in scenarios where clients are subject to data collection constraints; e.g., when clients can only participate in training intermittently, resulting in independent samples between consecutive participation rounds. In this paper, through a simple experiment, we demonstrate that such independence can improve convergence, especially for slow-mixing Markov processes.

2 Problem setup

2.1 Streaming Federated Learning

Consider M clients, each with an objective given by

$$F_m(w) := \mathbb{E}_{x\sim\pi_m}[f_m(w;x)], \tag{1}$$

where $f_m : \mathbb{R}^d \times \Omega_m \to \mathbb{R}$ is a smooth function on $\mathbb{R}^d$ and π_m is the data distribution of client m.

In FL, the M clients collaborate to solve

$$\min_{w\in\mathbb{R}^d} \frac{1}{M}\sum_{m=1}^{M} F_m(w) \tag{2}$$

Two popular examples of this procedure are Minibatch SGD and Local SGD [11]. In Minibatch SGD, T steps of SGD are performed, each based on a gradient computed from a minibatch of KM samples. To summarize, initializing at some w_0, Minibatch SGD operates as follows,

$$g_t^{(m,k)} = \nabla f_m\left(w_t; x_t^{(m,k)}\right), m \in [1,M], k \in [0,K-1],$$

$$w_{t+1} = w_t - \gamma g_t, \text{ where } g_t = \frac{1}{MK}\sum_{m=1}^{M}\sum_{k=0}^{K-1} g_t^{(m,k)}$$

Unlike Minibatch SGD, Local SGD allows the clients to update their local iterates throughout the local training phase based on their gradient estimates. Denote $w_t^{(m,k)}$ the local iterate after t rounds and k local steps on client m, Local SGD operates as follows,

$$w_t^{(m,k+1)} = w_t^{(m,k)} - \eta\nabla f_m\left(w_t^{(m,k)}; x_t^{(m,k)}\right),$$

$$w_{t+1}^{(m,0)} = w_{t+1} = \frac{1}{M}\sum_{m=1}^{M} w_t^{(m,K)}$$

In standard FL, the samples $(x_t^{(m,k)})_{k\in[K]}$ available at client m in any communication round t are fixed, corresponding to pre-collected data, while in streaming FL, the data available to client m can change in every communication round. This limits client control over sampling, where i.i.d. samples from the target distribution π_m are not available.

2.2 Markovian Data Streams

Let $x_t^m = (x_t^{(m,k)})_{k\in[K]}$ be the data samples available to client m in communication round t. Health and environmental monitoring data often follow non-stationary Markov processes, as seen in biological, chemical, and physical systems. As such, we model the data stream $X_m = (x_t^m,\ t \in \mathbb{N})$ of client m by a time-homogeneous Markov chain evolving on the state space $\Omega_m \subseteq \mathbb{R}^d$ with the corresponding Borel σ-field $\mathcal{B}_m$. The evolution of the time-homogeneous Markov chain for client m is characterized by the initial distribution $x_0^m \sim Q_m$ and by the transition kernel $P_m(x_t^{(m,k)}, \mathrm{d}x_t^{(m,k+1)})$. The probability of transitioning from the state $x_t^{(m,k)} \in \Omega_m$ to a state $x_t^{(m,k+1)} \in \mathcal{X}_m \in \mathcal{B}_m$ is given by $\int_{x\in\mathcal{X}_m} P_m(x_t^{(m,k)}, \mathrm{d}x)$. The t-step transition kernel for client m from an initial state x_0^m is denoted by $P_m^t(x_0^m, \mathrm{d}x)$. We focus on the case of independent clients, where X_m is independent of $X_{m'}$ for $m \neq m'$. This scenario occurs when client data streams are generated by non-interacting processes.

The Markov chain X_m admits a stationary distribution π_m on Ω_m if

$$\int_{x\in\Omega_m} \pi_m(\mathrm{d}x) P_m(x, \mathrm{d}x_t^{(m,k)}) = \pi_m(\mathrm{d}x_t^{(m,k)}). \tag{3}$$

We say that X_m is stationary if $x_0^m \sim \pi_m$; otherwise, X_m is non-stationary. In this paper, we focus on the stationary case (analysis in the non-stationary setting is given in [4]). The stationary distributions $\pi_m,\ m \in [M]$ correspond to the target distributions in (2), which capture the long-term statistics of the data samples.

As each client data stream evolves independently, the system-level data stream $X = ((x_t^m)_{m\in[M]},\ t \in \mathbb{N})$ will eventually draw samples from the stationary distribution π defined on $(\Omega := \bigtimes_{m=1}^{M}\Omega_m, \mathcal{B} := \bigotimes_{m=1}^{M}\mathcal{B}_m)$. Let $Q := \bigotimes_{m=1}^{M} Q_m$, $P(x_0, \mathrm{d}x) := \prod_{m=1}^{M} P_m(x_0^m, \mathrm{d}x^m)$, and denote the t-step transition kernel for the system-level Markov chain initially in state x_0 by $P^t(x_0, \mathrm{d}x)$.

The distance of the distribution of $(x_t^m)_{m\in[M]}$ and π can be quantified in terms of the total variation norm. The mixing time τ of the system-level data stream X is then defined as the required number of transitions so that this distance is sufficiently small (see [4] for further details). Following prior work on stochastic optimization with Markovian data [2], we assume that the system-level Markov chain admits exponential mixing, i.e., regardless of the initial distribution, it converges exponentially fast to the stationary distribution π.

In this paper, we consider the following scenarios:

(i) *Dependent Batches:* All TK samples observed by client m are drawn sequentially from the Markov chain X_m with transition kernel P_m and stationary initial distribution $\mathcal{Q} = \pi$.

(ii) *Independent Batches:* At the beginning of each communication round, the samples x_t^m are drawn from the Markov chain X_m with transition kernel P_m, and initial state $x_t^{m,1} \sim \pi_m$ drawn independently of x_{t-1}^m. As such, x_t^m is independent of x_{t-1}^m, while the samples $(x_t^{m,k})_{k\in[K]}$ remain statistically dependent.

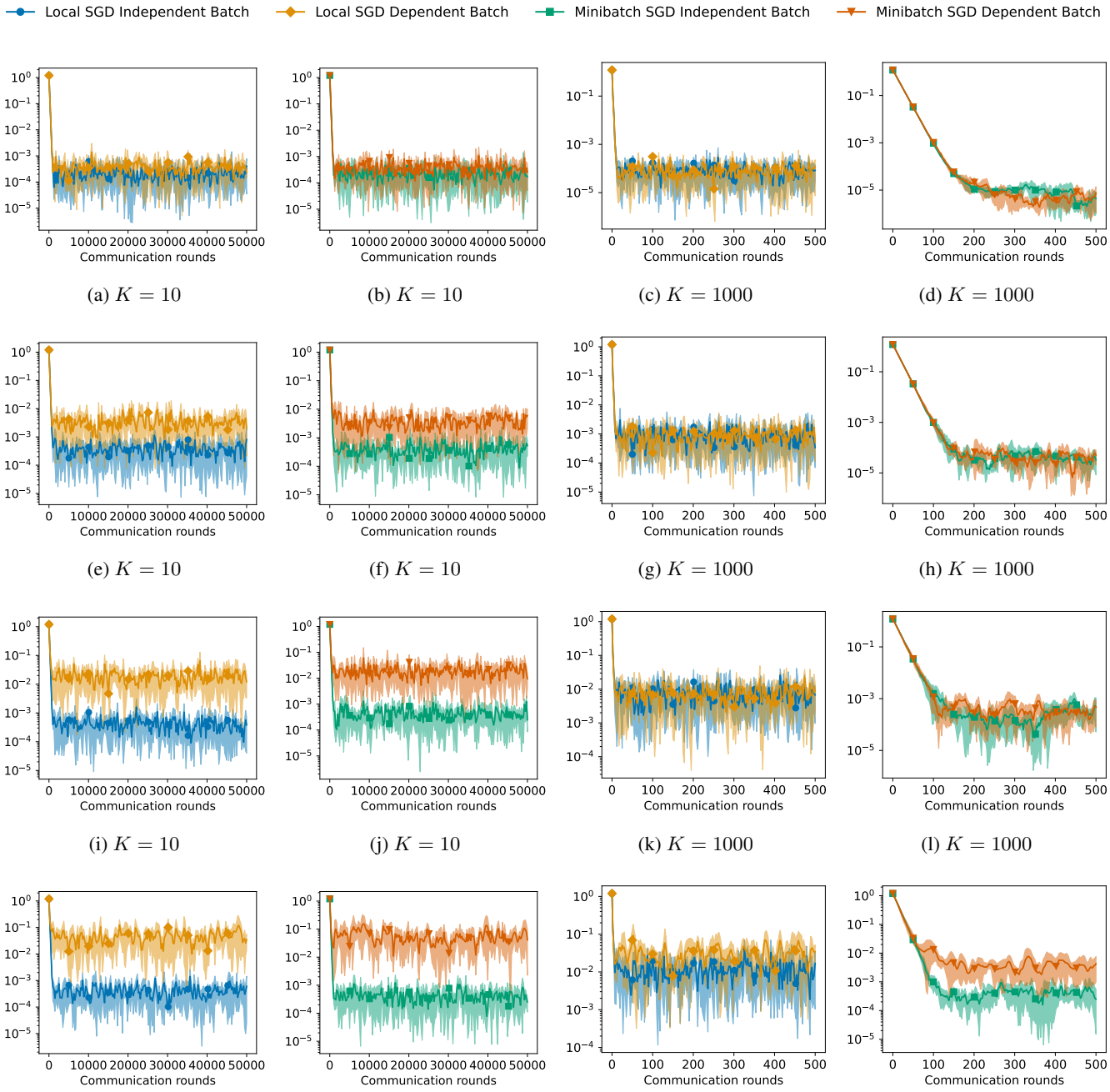
In [4], we focus on the first setting, although the theoretical analysis can also be applied to the second one. In the following section, we compare the performance of Minibatch SGD and Local SGD under these two settings through a simple toy experiment.

3 Results

In this section, we experimentally evaluate the performance of Minibatch SGD and Local SGD on a non-convex linear regression problem with Markovian data. In the following experiments, we study the average performance over 5 random seeds.

Let $\mathcal{U}(a,b)$ denote the uniform probability distribution over $[a,b]$. The data stream for each client m is generated by a two-state Markov chain $(i_t^{(m,k)},\ k \in [K], t \in \mathbb{N})$, where the probability of jumping from one state to another is $p \in (0,1)$, with mixing time $\Theta(1/p)$. [1] Associated with each state $i \in$

1. Here we use the standard Big-Θ notation.



(a) $K = 10$ (b) $K = 10$ (c) $K = 1000$ (d) $K = 1000$

(e) $K = 10$ (f) $K = 10$ (g) $K = 1000$ (h) $K = 1000$

(i) $K = 10$ (j) $K = 10$ (k) $K = 1000$ (l) $K = 1000$

(m) $K = 10$ (n) $K = 10$ (o) $K = 1000$ (p) $K = 1000$

Figure 1 – Gradient norm as a function of the number of communication rounds for Local SGD and Minibatch SGD with 100 clients, with mixing time $\tau_m = 10$ (first row), $\tau_m = 100$ (second row), $\tau_m = 1000$ (third row) and $\tau_m = 10000$ (last row) for all $m \in [M]$ and number of local steps $K \in [10, 1000]$.

$\{0, 1\}$ are the vectors $V_{m,i} \in \mathbb{R}^{10}$ which are drawn randomly for each seed according to $\mathcal{U}(0, 1)$. For each seed and $i \in \{0, 1\}$, half of the optimal parameters $w_{m,i} \in \mathbb{R}^{10}$ take the value $w_i^1 \sim \mathcal{U}(0, 1)$, while the other half take the value $w_i^2 \sim \mathcal{U}(1, 2)$.

The samples associated with client m correspond to the observations

$$x_t^{(m,k)} = w_{m,i_t^{(m,k)}}^T V_{m,i_t^{(m,k)}} + \epsilon_t^{(m,k)}, \tag{4}$$

where $\epsilon_t^{(m,k)} \sim \mathcal{U}(0, 0.01),\ t \in \mathbb{N}, m \in [M], k \in [K]$.

Since the Markov chain $(i_t^{(m,k)},\ k \in [K], t \in \mathbb{N})$ is symmetric, the stationary distribution for every client m is uniform; i.e., $\pi_m(0) = \pi_m(1) = 1/2$. The local objective function for each client m is then given by

$$\begin{aligned} F_m(w) =& \frac{1}{2}\left[(w - w_{m,1})^T V_{m,1}\right]^2 \\ &+ \frac{1}{2}\left[(w - w_{m,2})^T V_{m,2}\right]^2 + \lambda r(w), \end{aligned} \tag{5}$$

where $r(w) = \frac{1}{2}\sum_{d=1}^{10} \frac{w_{[d]}^2}{1+w_{[d]}^2}$ is a non-convex regularizer often used in robust non-convex and smooth stochastic optimization.

In Figure 1, we plot the gradient norm trajectory $\|\nabla F(w_t)\|^2$ of Minibatch SGD and Local SGD, with algo-

rithm parameters $\gamma = 0.01, \eta = 0.001$, $\beta = 0.1$, 100 clients, the number of local steps $K \in \{10, 1000\}$, and the mixing time for every client $\tau_m \in [10, 100, 1000, 10000]$, under two scenarios described in Section 2.2. The curves indicate that when the mixing time and the number of local steps are of the same order of magnitude, both algorithms perform similarly in both scenarios. However, when the mixing time of the underlying Markov processes is significantly larger than the number of local steps, ensuring independence between samples across different communication rounds can improve convergence. This suggests that the delays in updating clients' memory caches due to communication overheads can actually be beneficial in FL systems with slow-mixing Markovian data streams. Therefore, designing suitable memory management techniques on the client side that account for these data collection constraints to maintain independence between samples across rounds could be an interesting direction to explore.

3.1 Theoretical Explanation

We provide a brief theoretical explanation of the above phenomena for Minibatch SGD. Detailed analysis can be found in our previous work [4].

The difference in performance between the two settings arises due to errors in stochastic gradient estimates. In the *Dependent Batches* case, we have that:

$$\mathbb{E}\left[\|g_t - \nabla f(w_t)\|^2\right] = \mathcal{O}\left(\frac{C_\infty \sigma^2}{\nu_{ps} MK}\right), \tag{6}$$

$$C_\infty = \sup_{x=(x_m)_{m=1}^M} \left\| \frac{dP(x,.)}{d\pi(.)} \right\|_{\pi,\infty} \tag{7}$$

and ν_{ps} is the *pseudo spectral gap* of P [9]. On the other hand, in the case of *Independent Batches*, we have:

$$\mathbb{E}\left[\|g_t - \nabla f(w_t)\|^2\right] = \mathcal{O}\left(\frac{\sigma^2}{\nu_{ps} MK}\right). \tag{8}$$

For product Markov chains that are uniformly ergodic (i.e., exponential convergence to the stationary distribution),

$$C_\infty = \mathcal{O}\left(\rho^{\lfloor 1/\tau \rfloor}\right), \ \rho < 1. \tag{9}$$

As such, C_∞ increases with the mixing time τ of the product Markov chain.

4 Conclusion

Statistical dependence in data streams is often unavoidable, making it crucial to study its impact on FL, especially as it emerges as a standard framework for collaborative learning between edge devices. This work compares the performance of FL algorithms under two relevant settings, both incorporating such dependence: one where client data is sampled consecutively from a Markov chain; and another where samples across communication rounds are independent while remaining dependent within the same round. Our experiments show that ensuring independence across rounds significantly improves convergence, particularly with slow-mixing Markov processes.

5 Acknowledgement

This research was supported in part by Groupe La Poste, sponsor of the Inria Foundation, in the framework of the FedMalin Inria Challenge, the LearnNet Inria Challenge, and the ANR JCJC TCDTP. Experiments were carried out using the Grid'5000 testbed supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations.

References

[1] D. Anderson and T. Kurtz. *Stochastic Analysis of Biochemical Systems*. Springer, Berlin, 2015.

[2] Aleksandr Beznosikov et al. First order methods with markovian noise: from acceleration to variational inequalities. In *Advances in Neural Information Processing Systems*, 2023.

[3] Thinh Doan et al. Finite-time analysis of distributed TD(0) with linear function approximation on multi-agent reinforcement learning. In *International Conference on Machine Learning*, 2019.

[4] Tan-Khiem Huynh, Malcolm Egan, Giovanni Neglia, and Jean-Marie Gorce. Streaming federated learning with markovian data. *arXiv preprint*, arXiv:2503.18807, 2025.

[5] Peter Kairouz et al. Advances and open problems in federated learning. *CoRR*, abs/1912.04977, 2019.

[6] Sajad Khodadadian et al. Federated reinforcement learning: Linear speedup under Markovian sampling. In *International Conference on Machine Learning*, 2022.

[7] Paul Mangold et al. SCAFFLSA: Taming heterogeneity in federated linear stochastic approximation and TD learning. In *Annual Conference on Neural Information Processing Systems*, 2024.

[8] O. Marfoq, G. Neglia, L. Kameni, and R. Vidal. Federated learning for data streams. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

[9] Daniel Paulin. Concentration inequalities for Markov chains by Marton couplings and spectral methods. *Electronic Journal of Probability*, 20(none):1 – 32, 2015.

[10] Han Wang, Aritra Mitra, Hamed Hassani, George J. Pappas, and James Anderson. Federated TD learning with linear function approximation under environmental heterogeneity. *Transactions on Machine Learning Research*, 2024.

[11] Blake E Woodworth, Kumar Kshitij Patel, and Nati Srebro. Minibatch vs local sgd for heterogeneous distributed learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 6281–6292, 2020.

[12] Q. Yuan, H. Shen, T. Li, Z. Li, S. Li, Y. Jiang, H. Xu, W. Tan, Q. Yang, J. gg, and J. Gao. Deep learning in environmental remote sensing: Achievements and challenges. *Remote sensing of Environment*, 241, 2020.