Data Efficient Sequential Decision Making in High Dimensions

by

Vineet Sunil Gattani

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved November 2024 by the
Graduate Supervisory Committee:

Gautam Dasarathy, Chair
Visar Berisha
Nicolò Michelusi
Giulia Pedrielli

ARIZONA STATE UNIVERSITY

December 2024

ABSTRACT

As machine learning (ML) systems rapidly advance, their scale and data requirements have surged, increasing the need for efficient data use while maintaining high performance and accuracy. This dissertation addresses the challenge of data efficiency in large-scale machine learning, particularly in sequential decision-making (SDM) problems. Numerous modern applications, from drug discovery to robotics to online recommendation systems, can be framed as SDM problems. While many frameworks exist for addressing SDM, this work focuses on two key paradigms: Federated Learning (FL) and Stochastic Bandits (SB).

In both FL and SB, an agent learns iteratively by observing data, making decisions, and refining those decisions to increase cumulative reward. This dissertation aims to reduce the data exchanged per interaction round between the agent and the environment and to minimize the total data required to achieve optimal model performance. The main contributions include: first, a communication-efficient FL methodology to address bandwidth limitations, noisy communication, and client data heterogeneity, ensuring robust performance; and second, a partial client participation strategy that enhances data efficiency in large, distributed user settings. For SB, a method is introduced to leverage low-dimensional structures in high-dimensional, partially observed data, enabling effective learning despite incomplete information. Additionally, SB is extended to accommodate periodically changing reward patterns, adapting the model to non-stationary environments.

These contributions advance data-efficient SDM strategies for complex, distributed environments, enabling scalable systems that adapt to dynamic, real-world contexts.

DEDICATION

To my parents

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Gautam Dasarathy, for his invaluable guidance, encouragement, and support throughout this journey. His expertise and mentorship have been instrumental in shaping this work, and I am profoundly thankful for the countless discussions and insights that helped me grow as a researcher.

I would also like to extend my sincere appreciation to the members of my committee for their valuable feedback and thoughtful suggestions, which have significantly enhanced the quality of this research. Your time and effort in reviewing my work and providing critical insights have been immensely beneficial.

A heartfelt thank you goes out to my peers and friends, who have been a constant source of support and a sounding board for my ideas. Your willingness to engage in discussions and offer feedback has been an incredible help.

I am deeply grateful to my family, for their unwavering support, encouragement, and understanding. Their belief in me has been a driving force throughout this process.

To my wife Shea, my pillar of strength, thank you for your endless love, patience, and support. You have been by my side through every step of this journey, taking care of our son and giving me the space and time I needed to complete this work. I couldn't have done this without you.

Lastly, I would like to express my gratitude to Jeffree, our adorable dachshund, who has been a constant companion at my desk, offering silent support during meetings and writing sessions.

Thank you all for your unwavering support and for making this journey possible.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Chapter 1

INTRODUCTION

In recent years, the field of machine learning (ML) has experienced unprecedented growth, driving transformative advancements across various sectors of technology and industry. As ML systems continue to evolve, they encounter new challenges, particularly those related to scalability and the vast amounts of data required for training effective models. This dissertation addresses a critical and timely issue in modern machine learning: **data efficiency** in large-scale systems, with a particular emphasis on **sequential decision-making (SDM)** problems.

Sequential decision-making is a fundamental framework in machine learning, where an agent interacts with an environment, observes outcomes, and makes decisions based on these observations. The objective in SDM is for the agent to learn a strategy that maximizes cumulative reward by making informed decisions over time. This iterative process of learning and adaptation is analogous to how humans learn and adapt to new experiences, making SDM a compelling and flexible framework to model a wide range of real-world applications.

For instance, in clinical trials, the agent could represent a treatment allocation system that learns to recommend therapies yielding the highest patient recovery rates. In robotics, the agent could be a robot navigating an environment, choosing actions that maximize efficiency or minimize task completion time. In distributed machine learning systems, the agent represents a central coordinator that aggregates information from multiple clients, each possessing a subset of the overall data. Here,

the goal is to learn a model that generalizes well across diverse data sources, maximizing prediction accuracy or minimizing a predefined loss function.

The challenges inherent in the SDM framework become significantly more pronounced in high-dimensional settings and when data sharing incurs high costs. Each interaction between the learning agent and the environment—whether it involves collecting new data, updating model parameters, or transmitting information—can be resource-intensive. As the scale of the system increases, the data requirements and associated costs also escalate. This dissertation tackles these issues by focusing on two key aspects: **communication efficiency** and **data efficiency**.

- **Communication efficiency** pertains to minimizing the overhead involved in exchanging data across distributed systems, particularly under bandwidth constraints. This is crucial in scenarios where multiple clients or devices must collaboratively train a model without the luxury of extensive data transfer, such as in federated learning systems operating over low-bandwidth networks.
- **Data efficiency**, on the other hand, emphasizes reducing the number of samples required for effective inference and learning. This aspect is critical in situations where obtaining labeled data is expensive or time-consuming, such as in medical diagnosis or personalized recommendation systems.

In this dissertation, we develop robust and accurate machine learning models that address these challenges within two paradigms: **Federated Learning (FL)** and **Stochastic Bandits (SB)**. Both FL and SB can be classified under the broader category of sequential decision making, where models are updated iteratively based on new observations or feedback. The FL framework is motivated by the need to reduce communication overhead while simultaneously learning model parameters across multiple clients. In contrast, SB focuses on efficiently utilizing large-scale data

for learning, particularly in dynamic environments where sequential decision-making is paramount.

In recent years, Federated Learning has emerged as an important paradigm for training high-dimensional machine learning models when training data is distributed across several edge devices such as smartphones and IoT sensors. However, training over wireless channels introduces several challenges, including bandwidth limitations, unreliability, noise in communication channels, and statistical heterogeneity in data across edge devices. The need to communicate model parameters from edge devices to the central server can become a significant bottleneck, particularly as real-world datasets and model parameters scale to millions. Inevitably, noise during transmission can induce bias in learning global model parameters. Furthermore, heterogeneous data across edge devices complicates the training process; if not managed properly, it can significantly prolong training time and result in poor model performance. Therefore, it is crucial to design FL algorithms that can handle heterogeneous data distributions and reduce communication costs effectively. While there have been siloed efforts investigating these fundamental challenges separately, this dissertation proposes a holistic approach that integrates these concerns, which will be discussed in Chapter 2.

Additionally, traditional FL algorithms often assume full client participation. In scenarios with limited communication overhead, this assumption may not hold, necessitating a shift towards partial client participation and an efficient querying scheme. To address this, the dissertation explores an active learning-based querying method for partial client participation, demonstrating its effectiveness in achieving convergence despite constraints, as outlined in Chapter 3.

The other key paradigm in the sequential decision-making landscape is Stochastic Linear Bandits (SLB), a powerful extension of the traditional Stochastic Bandits

framework. While standard SB models typically consider a finite set of actions with unknown reward distributions, SLB incorporates a richer set of features for each action, allowing for a more nuanced decision-making process. In SLB, the agent operates over a series of rounds, where each action is represented by a feature vector, and the expected reward is assumed to be a linear function of these features. As datasets grow in size and complexity, the challenge of high-dimensional feature representations in recommendation systems becomes increasingly pronounced.

Moreover, the increasing emphasis on privacy protection complicates the process of feature representation in Stochastic Linear Bandits (SLB), as sensitive user characteristics may result in incomplete or partially observed feature vectors. In addition to privacy concerns, the high dimensionality of feature vectors often necessitates strategies to reduce communication costs. One common approach is to transmit sparse feature vectors to the learning agent, effectively reducing the overhead while preserving essential information. Alternatively, a noisy transmission model can be assumed, where the noise naturally sparsifies the high-dimensional feature vector during communication.

However, in practice, not all features contribute equally to the recommendation task. This observation highlights a key property often present in SLB problems: an underlying low-dimensional structure in the feature space. Exploiting this low-dimensional structure is crucial for enhancing learning efficiency, as it allows the agent to focus on the most informative components of the feature vector, thereby reducing the sample complexity required for effective learning.

Recognizing this, Chapter 3 introduces a novel methodology that addresses the challenge of learning in high-dimensional settings with missing observations. By efficiently estimating the underlying low-dimensional subspace, our approach enables

sample-efficient learning even when only a subset of the features is observed or transmitted. This subspace estimation technique not only improves the performance of SLB methods but also enhances their robustness to missing data, making it a powerful tool for real-world applications where complete feature information is rarely available.

Lastly, in the traditional Stochastic Bandits (SB) setting, the learning agent typically selects from a finite set of actions with unknown but stationary reward distributions. However, an intriguing challenge arises when the reward distributions are non-stationary—exhibiting changes over time that follow a structured pattern, such as periodic fluctuations. Non-stationarity is common in many real-world applications, where underlying trends or seasonal effects cause shifts in the reward distribution.

For example, consider the task of selecting the most profitable stock from a set of options. Stock prices often exhibit periodic patterns due to factors like quarterly earnings reports, market cycles, or seasonal trends. In such cases, the goal is to identify the stock that maximizes returns while accounting for these recurring patterns. The introduction of non-stationary rewards complicates the learning process, as erratic or unpredictable changes can mislead standard SB algorithms designed for stationary settings.

To address this, we propose a novel methodology that can adapt to non-stationary environments by learning the underlying periodic patterns in a data-efficient and robust manner. This approach allows the agent to make optimal decisions even in the presence of temporal shifts or uncertainties. The details of this methodology are discussed in Chapter 4.

Collectively, these contributions enhance the practicality and reliability of ML systems in resource-constrained environments, offering solutions that are both theoreti-

cally grounded and applicable to real-world scenarios. By emphasizing communication and data efficiency, this work paves the way for more robust and adaptable ML systems across various domains, addressing critical challenges in data exchange, decision-making with limited information, and rapid performance assessment in dynamic, real-world settings.

Chapter 2

COMMUNICATION EFFICIENT FEDERATED LEARNING OVER BANDLIMITED NETWORKS

2.1 Introduction

In recent years, federated learning has emerged as an important paradigm for training high-dimensional machine learning models when the training data is distributed across several edge devices. However, when training is carried out over wireless channels in a federated setting, a number of challenges arise, including bandwidth limitations, unreliability and noise in communication channels, and statistical heterogeneity (non-identical distribution) in data across edge devices Kairouz et al. 2021. In what follows, three key challenges are elaborated upon. Firstly, with the size of real-world datasets and the machine learning model parameters scaling to the order of millions, communicating model parameters from edge devices to the server and back can become a major bottleneck in model training if not handled efficiently. Needless to say, the transmission of model parameters to the central server over wireless channels is noisy and unreliable in nature. In practice, channel noise is inevitable during the training process and will induce bias in learning the global model parameters. Furthermore, the data collected and stored across edge devices is heterogeneous, which adds an extra layer of complexity due to diversity in local gradient updates. If statistical heterogeneity across edge devices is not handled properly, it can significantly extend the training time and cause the global model to diverge, resulting in poor and unstable performance. Thus, designing Federated Learning (FL) algorithms that can effectively handle heterogeneous data distributions while minimizing communication overhead is

of critical importance. While previous research has largely focused on addressing these fundamental challenges in isolation, a comprehensive approach—Federated Proximal Sketching (FPS)— is proposed which integrates solutions to both challenges, providing a unified framework that enhances FL performance in practical, real-world settings.

To address the first key challenge of communication bottleneck, the use of count sketch (CS) Charikar et al. 2002 as an efficient compression operator for model parameters is proposed, as illustrated in Figure 1. The CS data structure is not only easy to implement but also comes with strong theoretical guarantees on the recovery of significant coordinates or heavy hitters. The CS data structure also enables us to apply the gradient updates easily so that at every time instant, we preserve information about the most important model parameters.

With such a compressed representation of the model parameters, over-the-air computing is employed to aggregate local information transmitted by each device. Specifically, over-the-air Abari et al. 2016; Goldenbaum et al. 2013 takes advantage of the superposition property of wireless multiple access channels, thereby scaling signal-to-noise ratio (SNR) well with an increasing number of edge devices.

To tackle the challenges due to data heterogeneity, the proximal gradient method is applied to 'reshape' the loss function by adding a regularization term. The regularization term is carefully designed such that it keeps the learned model parameters from diverging in the presence of data heterogeneity. Additionally, using experimental studies I demonstrate empirically that this modification to our loss function helps us reduce the number of communication rounds to the central server while maintaining high accuracy.

The regularization term also helps in curbing the effect of noise due to communication over wireless channels. There is an interesting line of literature highlighted in the

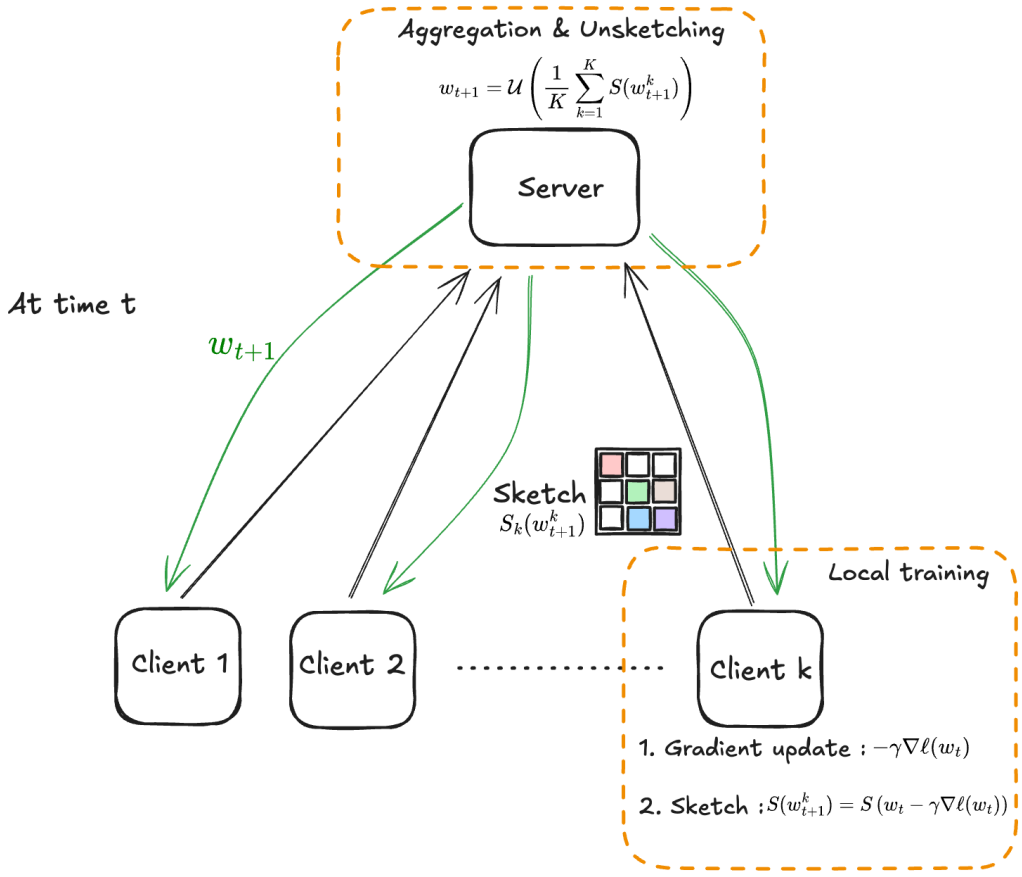


Figure 1. Illustration of Federated Proximal Sketching (FPS) over wireless multi-access channel (MAC).

Section 2.2.2, which studied learning in the presence of noise by using regularization. In addition, the count sketch data structure is used to produce reliable estimates of the "heavy hitter" (i.e., salient) coordinates. As the count sketch data structure uses multiple hashing functions, the process of sketching and unsketching provides denoising effect, and further the randomized nature of the hash functions produces a noise robust estimates of top-k coordinates. The usage of count sketching in conjunction with regularization forms the core of our strategy to tackle the challenge of FL under noisy wireless channel settings.

The main contributions of this paper can be summarized as follows:

- **Federated Proximal Sketching.** The proposed Federated Proximal Sketching (FPS), is a novel and robust count-sketch based algorithm for federated learning in noisy wireless environments. FPS is designed to be highly communication-efficient and can effectively handle high-level data heterogeneity across edge devices.
- **Impact of Gradient Estimation Errors.** Because the communications of gradient updates over noisy wireless channels may result in bias. As a consequence a general biased stochastic gradient structure is considered and the impact of gradient estimation errors (including bias) is quantified. In the presence of biased gradient updates, the FPS algorithm converges with high probability to a neighborhood of the desired global minimum, where the size of the neighborhood hinges upon the bias induced, under mild assumptions. Note that the biased stochastic gradient structure here is more general than the existing line of works on FL Stich et al. 2018; Ivkin et al. 2019; Karimireddy et al. 2020, which do not address the bias in the stochastic gradients, a key aspect in a large number of practical problems.
- **Statistical Heterogeneity.** Theoretically investigations explore the impact of varying degrees of statistical heterogeneity in data distributed across devices on the convergence. The comprehensive study is motivated by T. Li et al. 2020 to tackle data heterogeneity and extends it to the bandlimited noisy wireless channel setting. A key insight that emerges from our analysis of FPS is that an interplay exists between the degree of data heterogeneity, rate of convergence, and choice of learning rate.
- **Experimental Studies.** The theoretical studies are complemented by numerical experiments on both synthetic and real-world datasets. Our experimental

results unequivocally demonstrate that FPS exhibits robust performance under noisy and bandlimited channel conditions. To evaluate the performance of FPS algorithm under varying degrees of class imbalance across edge devices, different data partitioning strategies are investigated. The results show that, in practice, FPS achieves high compression rates on large-scale real-world datasets without significant loss in accuracy under different data distribution strategies. In some cases, improved accuracy of more than 10 - 40% over other competing FL algorithms in highly heterogeneous settings.

2.2 Related Work

This work looks at federated learning under three key challenges: (1) limited bandwidth across edge devices; (2) noisy wireless MACs; and (3) heterogeneous data distribution across devices. In what follows, I elaborate on different works which have addressed these three challenges until now.

2.2.1 Communication Efficient Federated Learning

Over the years communication-efficient stochastic gradient descent (SGD) techniques have been developed which reduce the cost of transmission using various gradient compression techniques like quantization Bernstein et al. 2018; Wu et al. 2018; Alistarh et al. 2017, sparsification Stich et al. 2018; Aji et al. 2017. Different sparsification methods like top$-k$ (in absolute value) and random$-k$ have been shown to converge in theory and empirical studies. However, such sparsification methods rely on the ability to store error accumulated by the compression scheme locally and re-introduce it in the next iteration to facilitate convergence Karimireddy et al. 2019. A major limitation of top$-k$ sparsification is the additional rounds of communication between

local edge devices to arrive at a consensus of global top$-k$ (heavy hitters) coordinates at each iteration. In a bandlimited setting where the number of edge of devices is large, this scheme is practically infeasible.

This work focuses on extending the current research on applying sketching as a compression scheme in federated learning. In Ivkin et al. 2019, a communication efficient SGD algorithm was proposed which uses sketches to compress the high-dimensional gradient vectors across each of the edge devices using a count sketch data structure. However, their algorithm involves a second round of communication between the edge devices and central server to aid the estimation of top$-k$ coordinates. In practice, the second round of communication is not always feasible due to latency issues and bandwidth limitations. In Rothchild et al. 2020 as well, the authors proposed an algorithm - FetchSGD, which used sketching as a compression operator and achieved convergence without the additional rounds of communication. However, an additional error accumulation count sketch data structure has to be maintained at the central server to facilitate convergence. In addition, their work claims that FetchSGD performs well when data is distributed in a non-IID manner across edge devices but fails to provide any algorithmic details on how it deals with heterogeneous data distribution. It also lacks a detailed theoretical and practical analysis of the algorithm in different data heterogeneity scenarios which is discussed in our study. While the work in Ivkin et al. 2019; Rothchild et al. 2020 aim to use sketches as a mere compression operator, this works is motivated by Aghazadeh, Spring, LeJeune, et al. 2018 which utilizes the count sketch data structure to perform SGD recursively and thus, eliminating the need to have any additional CS data structures for error accumulation. In short, the gradient updates are added in the CS data structure at every time step where they are aggregated with all the past gradient updates, leaving

us with an compressed representation of model parameters. The original work in Aghazadeh, Spring, LeJeune, et al. 2018 was implemented for a single device and this work extends this to a federated learning in a band-limited noisy wireless channel setting.

2.2.2 Federated Learning over Wireless Channels

In the previous section, the focus was on communication efficient FL under a noiseless channel setting. In practice, the transmission of gradient vectors over wireless channels to the central server is noisy and erroneous. As a consequence of transmission over noisy channels there is bias induced in gradient update vectors transmitted. The authors of Ang et al. 2020 consider regularization based optimization of the loss function to mitigate the bias induced by wireless communications. The motivation for regularization based method stems from the works of Graves 2011; Goodfellow et al. 2016, where training with noise was approximated via regularization to enhance the robustness of neural networks. Many regularizers are available, but no single one is universally superior for handling noise. Therefore, a problem-specific regularization term must be chosen. Due to its simplicity and ease of implementation, ℓ_2-regularization is selected.

To provide a holistic view of other related work of FL in wireless channel setting, an additional practical challenge considered is mitigating the effect of bias induced due to channel noise under limited power budget. Under such constraints, the authors in Zhang et al. 2021; Amiri et al. 2020 developed an adaptive power allocation strategy based on channel state information and magnitude of gradient vector coordinates to reduce the impact of communication error on convergence results (also see K. Yang et al. 2020; Zhu et al. 2019). While the above works considered only uplink channel

noise, more recently, in Wei et al. 2021, the authors analyzed the convergence of the well known FedAvg algorithm McMahan et al. 2016 under both noise in uplink and downlink transmission channels. While in this work, power constraints and any knowledge of channel state information are not considered, our work can be easily extended to a power constraint setting.

2.2.3 Statistical Heterogeneity across Edge Devices

One of the fundamental challenges in federated learning as stated in Section 2.1 is statistical heterogeneity in data across edge devices. Recent years have witnessed the development of algorithms, such as FedProx T. Li et al. 2020, FedNova J. Wang et al. 2020 and SCAFFOLD Karimireddy et al. 2020 to handle statistical heterogeneity. The algorithms listed above aim to reduce the drift of local iterates at each client from the global iterate maintained at the central server. The theoretical analysis of the convergence of the above-mentioned algorithms has also been well-studied under various assumptions that captures the dissimilarity in gradient computation across edge devices due to non-IID data distribution Kairouz et al. 2021. Here the bounded gradient dissimilarity assumption is used which is considered in T. Li et al. 2020 and it has been shown to be analogous to other commonly used dissimilarity assumptions like the bounded inter-client variance X. Li et al. 2021. However, these algorithms have not been studied in a band-limited and noisy wireless communication channel setting. The strategy used in FedProx is of particular interest to us, as it tackles the issue of statistical heterogeneity by appending a proximal term to the loss function.Building on this, later sections demonstrate that the proximal term in the algorithm serves two purposes: first, to reduce the effect of channel noise, and second, to aid convergence in the presence of statistical heterogeneity.

On a more practical side, recently a survey Q. Li et al. 2021 carried out an extensive experimental study on the above state-of-the-art algorithms over different data partitioning strategies and datasets. A particular kind of data partitioning strategy which is of interest to us is the label distribution skewness. A motivating example can be that some hospitals are specialized in certain kind of diseases and have data specific to it. An extreme case of label distribution skewness is where edge devices have access to only a few classes of labels Yu et al. 2020. Other notion of label skewness which is referred to as class imbalance in modern machine learning literature was studied in J. Wang et al. 2020; H. Wang et al. 2020; Yurochkin et al. 2019. Different degrees of statistical heterogeneity are simulated in the experimental section by varying the amount of class imbalance present at each edge device.

Furthermore, investigating these challenges mentioned above (bandlimitedness, statisitical heterogeneity and noisy channels) allows the work in this dissertation to uniquely sit at the intersection of analyzing and tackling the three key FL challenges specified above.

2.2.4 Bias in Stochastic Gradient Descent

The vanilla stochastic gradient descent has been well studied in presence of unbiased gradient updates Bottou et al. 2018. Recently, biased gradient updates have been considered in SGD, for instance, in large-scale machine learning systems techniques sparsification, quantization have been used to mitigate the issue of communication bottleneck. Such compression techniques produce biased gradient updates. There is a growing line of work on how different error accumulation and feedback schemes can mitigate the issue of bias and speed up convergence of SGD and distributed learning algorithms Karimireddy et al. 2019; Stich et al. 2018. More recent work on error

feedback can be found in Gorbunov et al. 2020; Qian et al. 2021.While this is not the focus of our paper, we are more interested in understanding how bias plays a role in theoretical convergence analysis of SGD. To this extent, we turn towards the body of literature that has dealt with modeling bias into the stochastic gradient structure. Our main motivation to have a more general stochastic gradient structure and mild conditions on bias and noise comes from the work in Ajalloeian et al. 2020a. Additional works that have considered similar assumptions are Stich 2019; Hu et al. 2021; Bottou 2010. Utilizing the assumptions from this line of work into distributed optimization literature (for this work, federated learning to be precise) will help analyze algorithms on a broader scale.

2.3 Preliminaries

2.3.1 Federated Learning over Wireless MACs

We begin with a federated learning setup where there are M edge devices and a central server. Only a fraction of the dataset $\mathcal{D}$ is available across each of the edge devices such that: $\mathcal{D} = \bigcup_{m=1}^{M} \mathcal{D}_m$. The loss function at an edge device m is defined as: $\ell_m(\mathbf{w}; \mathbf{x}_j, y_i)$, for a data sample $(\mathbf{x}_j, y_j) \in \mathcal{D}_m$. For a mini-batch ξ^m sampled at each device m, the loss function is denoted as:

$$f_m(\mathbf{w}; \xi^m) \triangleq \frac{\ell_m(\mathbf{w}; \xi^m)}{|\xi^m|}, \tag{2.1}$$

where, $|\cdot|$ represents cardinality of a set. The objective is to minimize the global loss function given by:

$$f(\mathbf{w}) := \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}_{\xi^m} \left[f_m(\mathbf{w}; \xi^m) \right]. \tag{2.2}$$

Here, the expectation is taken with respect to the random process that samples mini-batches at each edge device. This loss function is optimized iteratively to converge to

the optimal model parameter vector $\mathbf{w}^*$. At each edge device m and time step t, the stochastic gradient is computed using the sampled mini-batch ξ_t^m and represented as $\mathbf{g}_t^m(\mathbf{w}_t) := \nabla f_m(\mathbf{w}_t; \xi_t^m)$. Without loss of generality, the notation of $\mathbf{g}_t^m(\mathbf{w}_t)$ can be simplified to $\mathbf{g}_t^m$. The gradients are now transmitted over noisy multiple subcarriers via over-the-air protocol. The aggregated received gradient vector is defined as: $\mathbf{g}_t := \frac{1}{M}\sum_{m=1}^{M} \mathbf{g}_t^m + \mathbf{n}_t$. Here, $\mathbf{n}_t \in \mathbb{R}^d$ is the channel noise. The gradient descent update rule is carried out at the central server as:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \gamma \mathbf{g}_t, \tag{2.3}$$

where, γ is the fixed learning rate and $\mathbf{w}_{t+1}$ is model parameter vector. The updated iterate $\mathbf{w}_{t+1}$ is broadcasted back to all the edge devices. The computation of local stochastic gradients, transmission to the central server, and broadcast of the updated iterates are performed recursively until a small neighborhood around the global minimum $\mathbf{w}^*$ is reached. In general, transmission over wireless channels is noisy and the number of subcarriers are limited due to bandwidth constraints. As a consequence, the received gradient vector $\mathbf{g}_t$ is biased. Next, the count sketch compression operator and its recovery guarantees are discussed.

2.3.2 Count Sketch: A Probabilistic Data Structure for Frequency Estimation

In many data analysis and streaming applications, it's often necessary to estimate the frequencies of items in a large dataset or data stream. However, storing exact counts for every item can be prohibitively expensive in terms of memory, especially when dealing with high-dimensional data or limited resources. This is where probabilistic data structures like the count sketch come into play.

The count sketch, introduced by Charikar et al. 2002, is a space-efficient randomized algorithm for approximating item frequencies in streaming data. It provides a clever

solution to the frequency estimation problem by using hash functions to compress the input space into a much smaller sketch, while still maintaining the ability to recover frequency estimates with provable accuracy guarantees. Some of the key feature of Count Sketch are:

1. **Space efficiency:** It uses significantly less memory than storing exact counts for all items. For a vector of dimension d, Count Sketch uses $\mathcal{O}(\log d)$ memory for efficient storage.
2. **Update efficiency:** It supports fast updates to item frequencies in constant time.
3. **Query efficiency:** It allows for quick estimation of any item's frequency.

The count sketch is particularly useful in scenarios where:

1. The input data is too large to store in memory.
2. We need to track frequencies of items in a streaming fashion.
3. We are interested in identifying *heavy hitters* or frequent items in the data.

Now, let's delve into the structure and operation of the count sketch:

A count sketch S is a randomized data structure that keeps a matrix of buckets (or bins): $w \times b \sim \mathcal{O}(\log d)$, where b and w are chosen by the user to achieve certain accuracy guarantees. The count sketch algorithm uses w random hash functions h_j for $j \in [w]$ to map the vector's coordinates to buckets (or bins) b, $h_j : \{1, 2, \ldots, d\} \to \{1, 2 \ldots b\}$. In addition, the algorithm uses w random sign functions s_j for $j \in [w]$ as well that maps the coordinates of the vector randomly to $\{+1, -1\}$, $s_j : \{1, 2, \ldots d\} \to \{+1, -1\}$.

Consider a high-dimensional vector $\mathbf{g} \in \mathbb{R}^d$, then, the count sketch data structure S sketches the i^{th} coordinate of the vector $\mathbf{g}$ denoted as $\mathbf{g}(i)$, into the cell $S(j, h_j(i))$ by

incrementing the value of the cell by $s_j(i)\mathbf{g}(i)$. This is performed for every $j \in [w]$ and every coordinate $i \in [d]$. Originally, as count sketch was implemented in streaming data setting, for T updates to the vector $\mathbf{g}$, the count sketch data structure requires only $\mathcal{O}\left(\left(k + \frac{\|\mathbf{g}_{tail}\|_2^2}{\varepsilon^2 \mathbf{g}(k)}\right) \log dT\right)$ memory to provide unbiased estimate of the top-k or heavy hitter (HH) coordinates such that the following holds with high probability:

$$|\hat{\mathbf{g}}(i) - \mathbf{g}(i)| \leq \varepsilon \|\mathbf{g}\|_2, \forall i \in \text{HH}, \tag{2.4}$$

where, HH is the set of indices of heavy hitter or top-k coordinates. Details on the recovery guarantees of count sketch can be found in Charikar et al. 2002. For completeness, the main theoretical result on count sketch and it's recovery guarantees is stated here:

Theorem 1 (Count sketch) *For a vector $\mathbf{g} \in \mathbb{R}^d$, count sketch recovers the top-k coordinates with error $\pm\varepsilon\|\mathbf{g}\|_2$ with memory $\mathcal{O}\left(\left(k + \frac{\|\mathbf{g}^{tail}\|^2}{\varepsilon^2 \mathbf{g}(k)^2}\right) \log \frac{d}{\delta}\right)$; where $\|\mathbf{g}^{tail}\|^2 = \sum_{i \notin top-k}(\mathbf{g}(i))^2$ and $\mathbf{g}(k)$ is the k-th largest coordinate and this holds with probability at least $1 - \delta$.*

The proof of this theorem is deferred to Section 2.9.1.

It is important to observe caution that the vector being sketched (here, $\mathbf{g}$) should not have too many heavy hitter coordinates. If all the coordinates of a vector are heavy, the CS data structure will have coordinates colliding and the resulting unsketched vector would be error-prone.

Having established the utility of Count Sketch for efficient vector representation, attention is now turned to one of the core components of the proposed methodology: the *MISSION* subroutine.

2.3.3 MISSION Algorithm: Stochastic Gradient Descent using Count Sketch

The algorithm proposed in Aghazadeh, Spring, Lejeune, et al. 2018 is first initialized with a vector $\mathbf{w}_0$ and a count sketch data structure S with zero entries. At iteration t, the mini-batch stochastic gradient, $\mathbf{g}_t$, is computed using mini-batch ξ_t. The gradient update vector is formed by multiplying $\mathbf{g}_t$ with the learning rate, $-\gamma\mathbf{g}_t$, and the non-zero entries of this computed gradient update vector are added to the count sketch S. Next, MISSION extracts the top-k heavy hitters from the sketch, $\mathbf{w}_{t+1}$. The process of computing stochastic gradients and adding them to the sketch is repeated recursively until the desired number of iterations is reached or convergence occurs. This process is illustrated in Figure 2

Algorithm 1 MISSION

1: Initialize initial vector $\mathbf{w}_0$, Count Sketch S and learning rate γ
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Compute stochastic gradient using mini-batch ξ_t: $\mathbf{g}_t(\mathbf{w}_t)$
4: Sketch the local vector $(-\gamma\mathbf{g}_t)$ into $S(\mathbf{w}_t)$: $S(\mathbf{w}_t - \gamma\mathbf{g}_t)$
5: Unsketch and extract parameter vector: $\mathbf{w}_{t+1} = \mathcal{U}_k(S(\mathbf{w}_{t+1}))$
6: **end for**
7: Return: The top-k heavy-hitters of parameter vector $\mathbf{w}$ from the Count-Sketch

It is important to note that the MISSION algorithm was initially designed for application on a local device. The subsequent section elaborates on how this approach can be extended to a federated setup to achieve communication efficiency.

2.4 Proposed Method: Federated Proximal Sketching

The key steps of the FPS algorithm are outlined in Algorithm 2. In the following, the key steps are further elaborated.

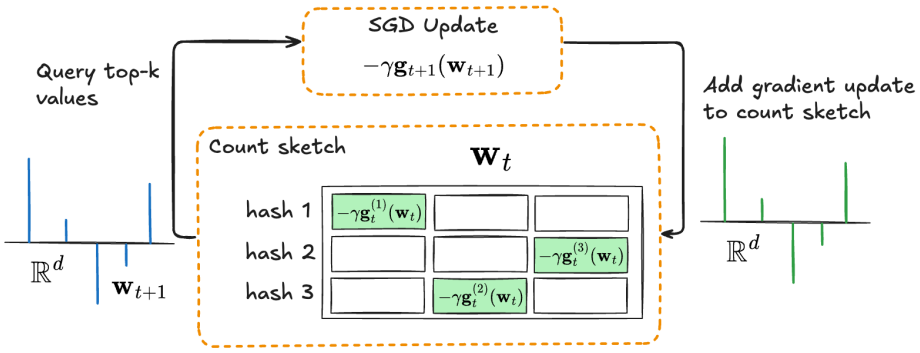


Figure 2. Illustration of one round of MISSION algorithm.

In Steps 1 and 2 of Algorithm 2, CS data structures at each of the edge devices and the central server are initialized to zero. Note that the size of the CS data structures is determined by the bandwidth available (number of subcarriers, K). I proceed with a fixed learning rate at each iteration. The number of local epochs/iterations E to be carried out before each global aggregation step is pre-determined. The appropriate choice of the number of local epochs is heuristic, and it is discussed in detail in Appendix 2.7.1.

In Steps 5 and 6 of Algorithm 2, the stochastic gradient is computed with respect to the mini-batch sampled at each edge device. The gradient update vector can be formed as: $-\gamma \mathbf{g}_t^m(\mathbf{w}_t^m)$ and sketch it into the CS data structure S^m maintained at that particular device m. To be more specific, sketching the gradient update vector to the CS data structure is implemented by the following mathematical operation in Step 6:

$$\begin{aligned}(-\gamma \mathbf{g}_t^m) \rightarrow S^m(\mathbf{w}_t^m) &\triangleq S^m(\mathbf{w}_t^m - \gamma \mathbf{g}_t^m(\mathbf{w}_t^m)) \\ &= S^m(\mathbf{w}_{t+1}^m).\end{aligned}$$

This is precisely the gradient update rule and implementation of this rule recursively is straightforward due to the linearity property of CS data structures. Observe that this update rule which compresses the computed gradient vector in a CS data

structure is reminiscent of the MISSION algorithm in Aghazadeh, Spring, Lejeune, et al. 2018. It is worth noting that MISSION was initially designed to operate on a single device, whereas FPS is a distributed algorithm where many instances of the MISSION algorithm are carried out in parallel. At every iteration in FPS, all edge devices maintain an efficient representation of the learned model parameter vector.

In Steps 8,9 and 10 of Algorithm 2, based on how frequently updates are pushed to the server, the CS data structure at each of the devices is transmitted over noisy wireless MAC channels. The received sketches are then aggregated. Top-k coordinate extraction is performed to obtain a k-sparse vector, $\mathbf{w}_{t+1}$, which is then broadcast back to the edge devices.

Steps 5-10 of Algorithm 2 are carried out recursively until convergence. As we are dealing with statistical heterogeneity across devices, aggregating updates after performing a set number of local updates helps. In cases where statistical heterogeneity is high, this strategy of performing local updates alone has been known to diverge empirically McMahan et al. 2016. This is illustrated in Figure 3. To address this issue, the loss function is restructured, and the advantages of this modification are discussed.

Loss function design. The restructuring follows the work in T. Li et al. 2020 with an added benefit of mitigating the effects of channel noise. The new loss function at each device is then given by:

$$f(\mathbf{w}, \mathbf{w}^{gb}) = \ell(\mathbf{w}) + \frac{\mu}{2} \left|\left|\mathbf{w} - \mathbf{w}^{gb}\right|\right|^2, \tag{2.5}$$

where, $\ell(\mathbf{w})$ is our application specific loss function, for instance, a cross-entropy loss for binary classification task or a mean-squared error for linear regression task. The iterate $\mathbf{w}^{gb}$ denotes the last aggregated model parameter vector that was broadcast

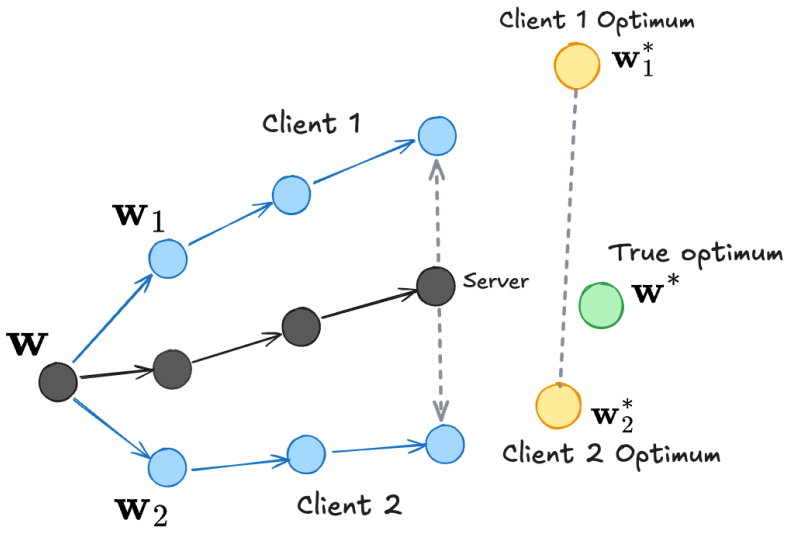


Figure 3. Illustration of client drift due to data heterogeneity across two edge devices.

by the central server. Therefore, for a non-zero proximal parameter μ, this new loss function provides the following benefits,

1. It controls the effect of statistical heterogeneity across devices by not letting the local updates $\mathbf{w}$ stray far away from the last global update $\mathbf{w}^{gb}$.
2. For improperly chosen number of local updates E, the proximal term minimizes the effect of divergence that would result as a consequence.
3. The ℓ_2 regularization (proximal term) applied to the global iterates induces a regularizing effect, constraining the magnitude of the iterates. As a result, the ℓ_2-norm of the global iterate, $\|\mathbf{w}^{gb}\|$, can be bounded above by some positive constant W, so that: $\|\mathbf{w}^{gb}\|^2 \leq W$.

2.5 Theoretical Analysis of Federated Proximal Sketching

As is standard, the loss function f_i at each edge device i is assumed to be L-smooth non-convex objective function.

Algorithm 2 Federated Proximal Sketching (FPS)

Inputs: Number of workers: M, mini-batches for each worker $m \in [M]$ at each time step: ξ_t^m, local epochs E.
Initialize individual sketches at each worker S^m with initial model parameters $\mathbf{w}_0^m$: $\mathbf{w}_0 \rightarrow S^m = S^m(\mathbf{w}_0)$.
for $t = 1, 2, \ldots, T$ **do**
    for $m = 1, 2, \ldots, M$ **do**
        Compute stochastic gradient using mini-batch ξ_t^m: $\mathbf{g}_t^m(\mathbf{w}_t^m)$.
        Sketch the gradient update vector $(-\gamma \mathbf{g}_t^m)$ at each worker: $(-\gamma \mathbf{g}_t^m) \rightarrow S^m(\mathbf{w}_t^m) = S^m(\mathbf{w}_{t+1}^m)$ and broadcast it to the central server after E local epochs.
    end for
    Receive aggregated sketches at the server: $S_t(\mathbf{w}_{t+1}) = \frac{1}{M} \sum_{m=1}^{M} S^m(\mathbf{w}_{t+1}^m) + n_t$.
    Unsketch and extract top-k coordinates of parameter vector: $\mathbf{w}_{t+1} = \mathcal{U}_k(S_t(\mathbf{w}_{t+1}))$.
    Broadcast k-sparse parameter vector to all edge devices: $\mathbf{w}_{t+1}^m = \mathbf{w}_{t+1}$.
end for

Assumption 1 ($L-$Smoothness) *A function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is $L-$smooth of for all $x, y \in \mathbb{R}^d$, it holds:*

$$|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| \leq \frac{L}{2} \|y - x\|^2. \tag{2.6}$$

In general, the received aggregate stochastic gradient $\mathbf{g}_t$, is biased, i.e., $(\mathbb{E}[\mathbf{g}_t] \neq \nabla f(\mathbf{w}_t))$, and this can be due to biased stochastic gradient estimation, data heterogeneity across devices and noisy channel conditions Zhang et al. 2021; Amiri et al. 2020. In what follows, the structure of stochastic gradient vector received at the central server is examined.

Definition 1 (Biased stochastic gradients) *Given a sequence of iterates $\{\mathbf{w}_t\}_{t=1}^T$, for all $t \in [T]$, the structure of biased stochastic gradient estimator can be written as:*

$$\mathbf{g}_t(\mathbf{w}_t) = \nabla f(\mathbf{w}_t) + \beta_t + \zeta_t, \tag{2.7}$$

where, β_t is the biased estimation error and ζ_t is the martingale difference noise. The

quantities β_t and ζ_t are defined as:

$$\beta_t := \mathbb{E}_t[\mathbf{g}_t(\mathbf{w}_t)] - \nabla f(\mathbf{w}_t) \tag{2.8}$$

$$\zeta_t := \mathbf{g}_t(\mathbf{w}_t) - \mathbb{E}_t[\mathbf{g}_t(\mathbf{w}_t)]. \tag{2.9}$$

Note that such a structure of stochastic gradient estimator has been studied in Zhang et al. 2008; Ajalloeian et al. 2020b. It directly follows from the above definition of bias and martingale difference noise that $\mathbb{E}[\zeta_t] = 0$. Here, the expectation $\mathbb{E}_t[\cdot]$ is with respect to ξ_t which is a realization of a random variable which represents the choice of single training sample in the case of vanilla SGD or may represent a set of sample in the case of mini-batch SGD, and the channel noise n_t. Furthermore, the bias and martingale noise terms satisfies the following assumptions.

Assumption 2 (Zero mean, $(P_n, \sigma^2)-$bounded noise) *There exists constants $P_n, \sigma^2 \geq 0$ such that:*

$$\mathbb{E}_t\left[\|\zeta_t\|^2\right] \leq P_n\|\nabla f(\mathbf{w}_t)\|^2 + \sigma^2. \tag{2.10}$$

Assumption 3 ($(P_b, b^2)-$bounded bias) *There exists constants $P_b \in (0, 1)$ and $b^2 \geq 0$ such that:*

$$\|\beta_t\|^2 \leq P_b\|\nabla f(\mathbf{w}_t)\|^2 + b^2. \tag{2.11}$$

These assumptions are significantly mild as the second moment bounds of the bias and noise terms scales with true gradient norm and constants b^2 and σ^2 respectively. By setting the tuple $(P_b, P_n, b^2, \sigma^2) = \bar{0}$, the case of unbiased gradient estimators is obtained. Convergence for this case has been well studied in literature.

Next, attention is turned to the compressibility of gradients. Specifically, it is assumed that the stochastic gradients are approximately sparse, as formalized in the following assumption HanQin Cai et al. 2022.

Assumption 4 (Approximately sparse gradients) *The stochastic gradients follow a power law distribution and there exists a* $p \in (1, \infty)$ *such that* $|\mathbf{g}_t(i)| = i^{-p}\|\mathbf{g}_t\|$.

In the Section 2.7.2, it is shown that some of the real-world datasets considered in this paper follow Assumption 4. As the value of p increases, it can be inferred that only a small number of coordinates in the vector $\mathbf{g}_t$ are significant. Therefore, by selecting an appropriate size for the count sketch (CS) data structure, efficient compression and strong recovery guarantees for the significant coordinates can be ensured.

Even though the loss functions across all the devices are same, as the data is distributed in a non-IID manner, due to random sampling of mini-batches across devices there will be dissimilarities in computation of loss functions and their respective gradient estimators. To this end, a measure of dissimilarity is defined between gradient estimators across edge devices similar to T. Li et al. 2020 as follows.

Definition 2 ($B-$local dissimilarity) *The local functions* f_m *are* $B-$*locally dissimilar at* $\mathbf{w}$ *if* $\|\mathbb{E}_{\xi_m}[\nabla f_m(\mathbf{w}; \xi_m)]\|^2 \leq \|\nabla f(\mathbf{w})\|^2 B^2$. *I further define* $B(\mathbf{w}) = \sqrt{\frac{\mathbb{E}_{\xi_m}[\|\nabla f_m(\mathbf{w};\xi_m)\|^2]}{\|\nabla f(\mathbf{w})\|^2}}$, *for* $\|\nabla f(\mathbf{w})\| \neq 0$.

Further, the following assumption ensures that the dissimilarity $B(\mathbf{w})$ defined in Definition 2 is uniformly bounded above.

Assumption 5 (Bounded dissimilarity) *For some* $\epsilon > 0$, *there exists* B *such that for all points* $\mathbf{w} \in S_\epsilon = \left\{\mathbf{w} \middle| \|\nabla f(\mathbf{w})\|^2 > \epsilon\right\}$, $B(\mathbf{w}) \leq B$.

If the data is distributed in an IID manner, the same loss function across all devices and the ability to sample an infinitely large sample size, then, $B \to 1$. However, due to different sampling strategies, in practice, $B > 1$. A larger value of B would

imply higher statistical heterogeneity across devices. Other formulations of measuring dissimilarity have been studied in Khaled et al. 2019; X. Li et al. 2019; J. Wang et al. 2019.

Let us denote $H = \frac{1}{1+2B^2(P_b+P_n)}$. Note that $H \leq 1$. I now define the following quantity $\rho(\gamma)$ as:

$$\rho(\gamma) \triangleq \frac{1 - P_b(1+2H)E^2B^2}{2} - \gamma(2 + 2P_bB^2 + (2(L+\mu)+1)P_nB^2)\,(1+2H)\,E^2, \tag{2.12}$$

where P_b, P_n, L and B are constants defined earlier; μ is the proximal parameter of our loss function and E is number of local epochs carried out at each edge device before global aggregation of model parameters at the central server. Let $f(\mathbf{w}^*)$ be the global minimum value of f. The range of values of the fixed learning rate γ considered are the ones that satisfy the following condition: $\rho(\gamma) \geq 1/4$ and is given by:

$$\gamma \leq \frac{1 - 6P_bE^2B^2}{12(1 + P_bB^2 + (L+\mu+1)P_nB^2)E^2}. \tag{2.13}$$

The CS data structure size I consider scales like $\mathcal{O}\left(ck\log\frac{dT}{\delta}\right)$. Here, c is some positive scalar ($c > 1$), k denotes the number of heavy hitter coordinates extracted or unsketched from the CS data structure, d is the ambient dimension, T is the number of iterations and δ is probability of error. The ℓ_2-norm of the iterates is bounded above by some arbitrary positive constant, $\|\mathbf{w}\|^2 \leq W$. The following main theorem pertains to the iterates in the FPS algorithm.

Theorem 2 *Under Assumptions 1, 2, 3, 4 and 5, the following result holds with*

probability at least $1-\delta$*:*

$$\frac{1}{T+1}\sum_{t=0}^{T}\|\nabla f(\mathbf{w}_t)\|^2 \leq \frac{|f(\mathbf{w}_0)-f(\mathbf{w}^*)|}{\gamma(T+1)} + \left(\frac{1}{c}+\frac{(k+1)^{1-2p}-d^{1-2p}}{2p-1}\right)W$$
$$+ 2E^2\left(1+2P_bB^2+\gamma(3+L+\mu+2P_bB^2+2P_nB^2)\right)b^2$$
$$+ 2E^2\left(1+\gamma(L+\mu+1)(3+2P_bB^2+2P_nB^2)\right)\sigma^2. \tag{2.14}$$

Remarks. A few important observations in order.

1. The first term on the right hand side of (2.14) is a scaled version of the term $|f(\mathbf{w}_0)-f(\mathbf{w}^*)|$, and its effect diminishes as $T\to\infty$.
2. The second term in (2.14) captures the error in unsketching top$-k$ coordinates of the iterates $\mathbf{w}_t$. It can be viewed as the residual error after extracting top$-k$ coordinates from the CS data structure. Assuming k is fixed, as the CS size increases, c increases and as a consequence $1/c$ is small in magnitude. The second quantity in this term depends on the value of k and p. As the sketch size increases, the ability to extract more heavy hitters increases, which implies that the value of k increases. Moreover, the value of p depends on the dataset. The magnitude of p depends on how effectively the relation between the input and the output can be represented using a small subset of features. The lesser the number of features used, the higher the value of p and vice versa. Thus, as the bandwidth at each edge device increases, the size of CS data structure increases as well and the effect of this term can be suppressed.
3. The third and fourth terms in (2.14) captures the effects of bias β_t and noise ζ_t, respectively. The iterates visit a neighborhood that scales by constants b^2 and σ^2 with high probability. Additionally, these terms scale as the values of local epochs E and the degree of heterogeneity B increase. To ensure the convergence of the algorithm, there is no fixed value of E that works for different

degrees of data heterogeneity. For example, as the data distribution across edge devices becomes increasingly heterogeneous, a smaller value of E can reduce the magnitude of the B^2 terms. This can be intuitively explained by arguing that as heterogeneity increases, choosing a higher value of local epochs will result in aggregating bias and noise due to the large dissimilarity in gradient computations across different edge devices. Therefore, as heterogeneity increases, the number of local epochs should be low to facilitate convergence, calling for more frequent communication with the central server.

4. Another aspect of our result arises from analyzing Equation (2.13), which provides a bound on the learning rate for facilitating convergence. It is noteworthy that if the dissimilarity B is large, a smaller learning should be selected. This intuitive approach makes sense because as the dissimilarity measure increases, the probability of local models diverging from the global minimum also increases. Hence, a smaller learning rate and fewer local epochs need to be chosen to stabilize the algorithm and ensure $\rho(\gamma) > 0$. Additionally, choosing a smaller learning rate can help reduce the size of the neighborhood scaled by bias and variance constants in the third and the fourth terms in Equation (2.14).

2.6 Experimental Studies

Several experiments are conducted on synthetic and real-world datasets with different model and environmental parameters. Under a bandlimited and noisy wireless channel setting, the performance of the proposed algorithm—FPS—is simulated alongside other competing bandlimited algorithms, such as FetchSGD Rothchild et al. 2020 and bandlimited coordinate descent (BLCD) Zhang et al. 2021. For the count sketch based algorithms like FetchSGD and FPS, the number of subcarriers or channels will

dictate the CS data structure size. In case of BLCD random sparsification is as a compression operator, therefore, the number of subcarriers will decide the number of coordinates of the gradient vector that will be selected at random for transmission to the central server. The number of edge devices M for all our experiments is chosen to be 10. The channel noise over each subcarrier follows a zero mean normal distribution, $\mathcal{N}(0, 1)$. For FetchSGD and BLCD, the global aggregation to the central server is performed at every epoch as designed in the papers they were proposed in. For FPS, the global aggregation is performed after every 5 local epochs. The number of local epochs is chosen heuristically and its choice is discussed more in the Appendix 2.7.1. A learning rate of 0.01 is chosen for all experiments. To simulate varying degrees of data heterogeneity, the following data partitioning scenarios are considered in our experiments:

Scenario 1. The data across all edge devices is distributed in an IID manner with equal number of samples corresponding to each class available.

The non-IID distribution considered in this work is label distribution skewness. Within this context, two sub-divisions of data partitioning strategies are identified: quantity-based label imbalance and distribution-based label imbalance.

Scenario 2. In this case, quantity-based label imbalance is considered, where, each edge device has access to samples corresponding to fixed number of classes only. For instance, in a binary classification problem the edge devices will have access to samples corresponding to only one class.

Scenario 3. Here, a distribution decides the proportion of samples of each label assigned to each edge device. A natural candidate for this task is a Dirichlet distribution. A hyperparameter α dictates how skewed the proportion of samples of each label across the devices will be. The probabilities are sampled according to

$p_l \sim \text{Dir}_M(\alpha)$ for a particular class label l. The probability vector p_l whose entries sum up to one, decides the proportion of samples of class l across all devices. Lower values of α correspond to highly skewed distribution of class labels and conversely, higher values correspond to a more even distribution of samples of each class across all devices. The value of α considered in this scenario is 0.1.

Scenario 4. In this case, the setup is the same as Scenario 3 with the value of hyperparameter for Dirichlet distribution set to $\alpha = 1$.

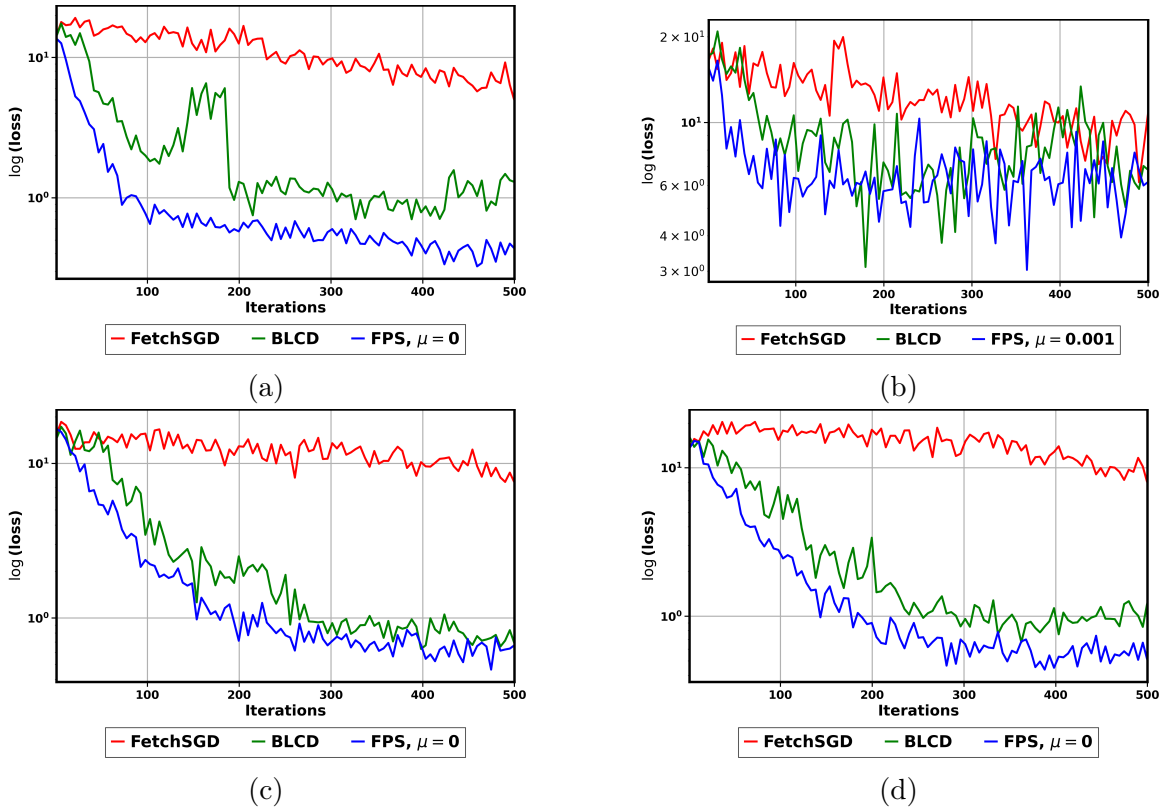


Figure 4. Plotting logarithm of test loss computed for FPS, BLCD, FetchSGD over 5 trials under noisy channel conditions with the gradients following Assumption 4 and power law degree $p = 5$. The figures correspond to different data partitioning strategies: (a) Scenario 1 (b) Scenario 2 (c) Scenario 3 (d) Scenario 4.

2.6.1 Synthetic Dataset

Data generation process. A regression task is considered in the synthetic data case. For scenario 1, consider generating observations, $\mathbf{y} = \mathbf{X}\mathbf{w} + 0.01\mathbf{n}$, where, $\mathbf{w} \in \mathbb{R}^d$ is the parameter vector, $\mathbf{n} \in \mathbb{R}^d$ is the additive Gaussian noise and whose each element n_i distributed according to $\mathcal{N}(0, 1)$. The design matrix is denoted by $\mathbf{X} \in \mathbb{R}^{N \times d}$ where each row $\mathbf{X}_i \in \mathbb{R}^d$ is a data sample distributed according to $\mathcal{N}(\bar{\mathbf{0}}, \Sigma)$. Here, the diagonal elements of Σ are non-zero and diminish such that $\Sigma_{ii} = i^{-p} \forall i \in [d]$.

For scenarios 2, 3 and 4, equal number of observations under two different distributions are generated, one where $\mathbf{X}_i \sim \mathcal{N}(\bar{\mathbf{0}}, \Sigma_1)$ and the other where $\mathbf{X}_i \sim \mathcal{N}(\bar{\mathbf{0}}, \Sigma_2)$. Here, $\Sigma_1 = \Sigma$ as defined in Scenario 1. The other diagonal matrix Σ_2 is chosen such that the diagonal elements are $\Sigma_{ii} = j^{-p}$, here, j is some random permutation of the index set $\{1, 2, \ldots, d\}$.

Experimental setup. The number of subcarriers allocated to each edge device are 256. For FPS, the set of values of proximal parameter considered are: $\mu = \{0, 0.001, 0.01, 0.1\}$. The ambient dimension d and power law degree p are set to 10000 and 5 respectively.

The average of the logarithm of test loss over 10 trials is plotted under a noisy bandlimited setting for FPS, FetchSGD, and BLCD in Figure 4. Starting from left to right, the figures correspond to data partitioning scenarios 1, 2 , 3 and 4 respectively. Across all experiments FPS achieves the lowest test loss. BLCD maintains comparable performance in Scenario 2 and a slightly weaker performance to FPS in other scenarios. Whereas FetchSGD exhibits poor performance across all scenarios. For each data partitioning scenario, the value of the proximal parameter for which FPS performs best is mentioned in the plot legends below.

2.6.2 Real-World Datasets

Apart from synthetic experiments, three real-word datasets are also considered, all of them corresponding to classification task. The average accuracy is reported corresponding to each of the data partitioning scenarios in noisy and noise-free case. The best choice of proximal parameter is selected from the set, $\mu = \{0, 0.01, 0.1, 1\}$ for each scenario is mentioned in the legend below each plot.

2.6.2.1 KDD12 – Click Prediction

The KDD12 dataset is binary classification task where the model must classify if a user will accept $\{1\}$ or reject $\{0\}$ an item recommended to it. Here, the items are news, games, advertisements, products. For more details on the dataset, see Juan et al. 2016. The number of features in this dataset are $54,686,452$. The number of subcarriers allocated to each edge device is, $K = 1024$.

In Figure 5, it is observed that FPS performs significantly better than FetchSGD and BLCD across all data partitioning strategies and noisy channel conditions. Additionally, FPS converges more quickly than other competing bandlimited algorithms. Table 1 reports the mean accuracy over 5 trials for various FL algorithms, including FPS, under varying degrees of statistical heterogeneity and channel noise conditions. In Figure 6, the performance of FPS, FetchSGD, and BLCD is plotted for different data partitioning strategies under noise-free channel conditions on the KDD12 dataset. When the data is distributed in an IID manner (scenario 1), FetchSGD performs slightly better than FPS. In scenario 2, where the data is highly heterogeneous, FPS outperforms other competing bandlimited algorithms. In scenarios 3 and 4, FPS matches the performance of FetchSGD.

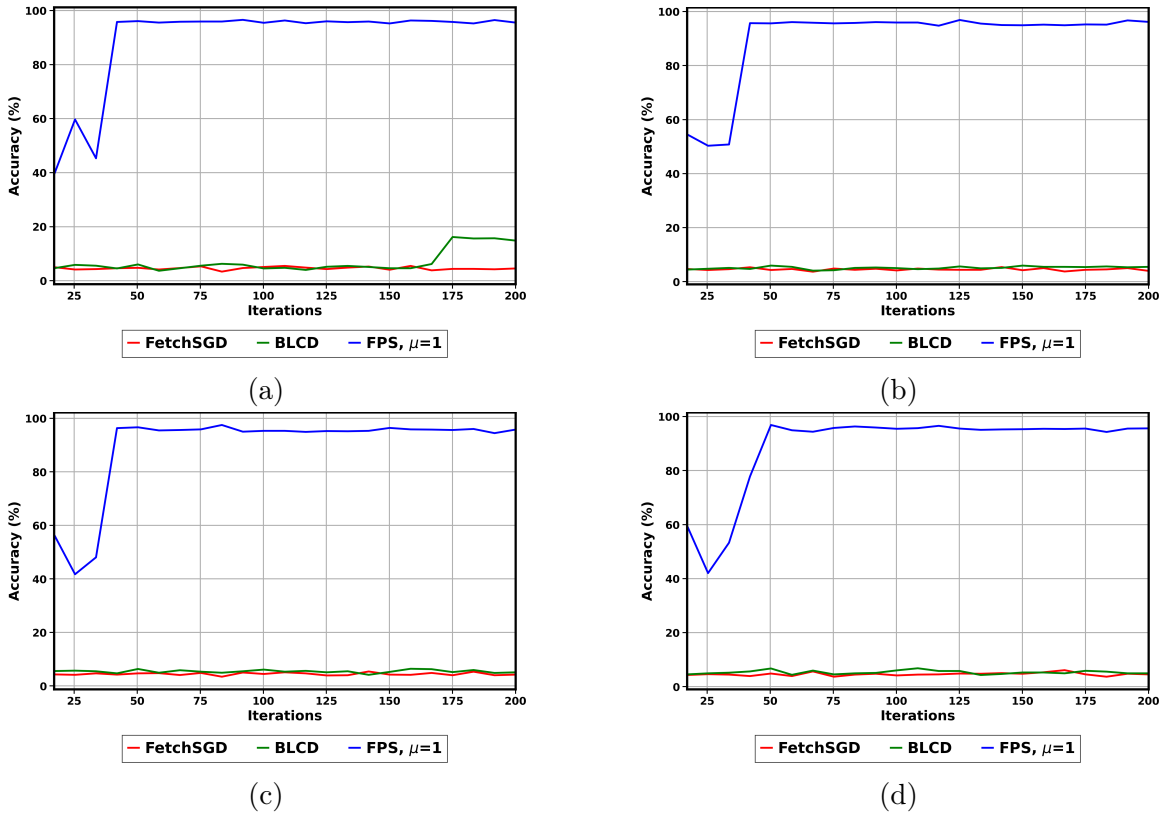


Figure 5. Plotting test accuracy for FPS, BLCD, FetchSGD on KDD12 dataset under noisy channel conditions. The figures correspond to different data partitioning strategies: (a) Scenario 1 (b) Scenario 2 (c) Scenario 3 (d) Scenario 4. I can observe that FPS converges to a global optimum quickly and outperforms other competing bandlimited algorithms by a huge margin.

2.6.2.2 KDD10 – Predicting Student Performance

The number of features in the dataset are $20,216,830$. For more details on the dataset, see Yu et al. 2010. The number of subcarriers that are allocated to each edge device, $K = 4096$.

In Figure 7, it is observe that FPS performs much better compared to FetchSGD and BLCD across all data partitioning strategies in bandlimited noisy channel conditions. Table 2, reports the mean accuracy over 5 trials for various FL algorithms including

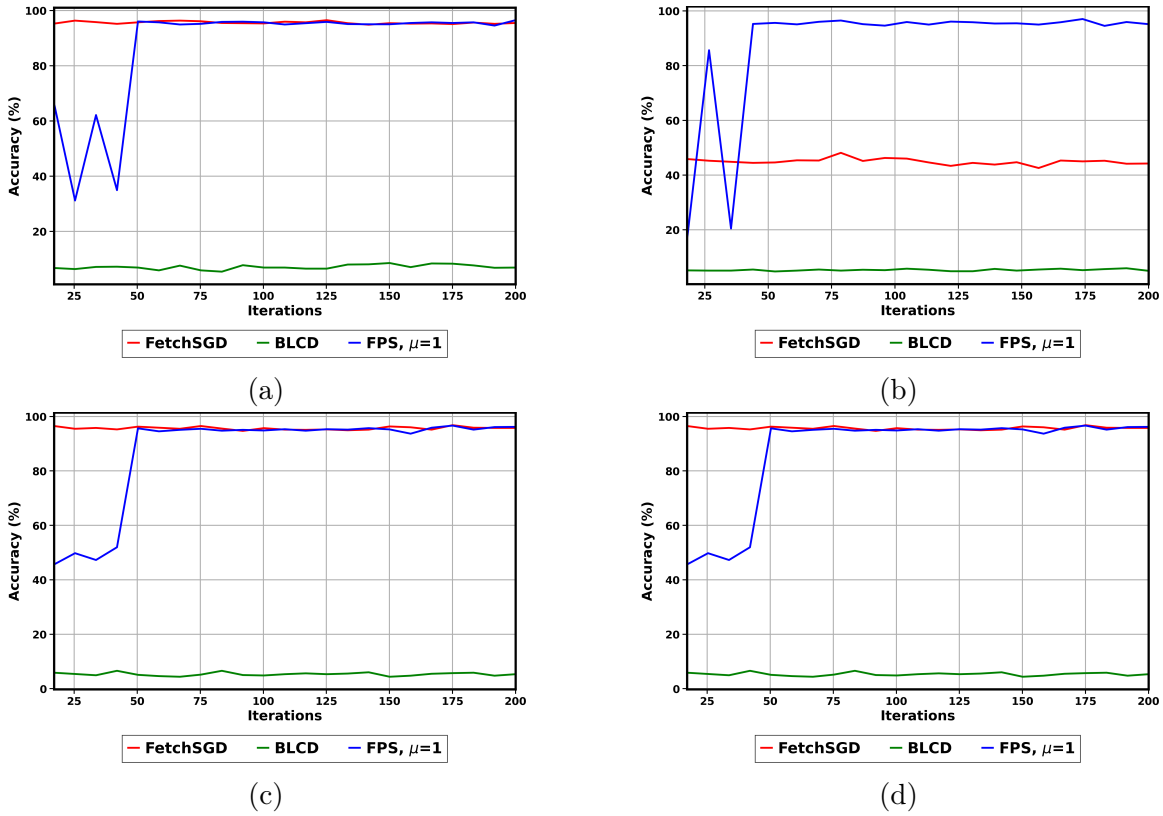


Figure 6. Plotting test accuracy for FPS, BLCD, FetchSGD on KDD12 dataset under noise-free channel conditions. The figures correspond to different data partitioning strategies: (a) Scenario 1 (b) Scenario 2 (c) Scenario 3 (d) Scenario 4.

FPS under varying degrees of statistical heterogeneity and channel noise conditions. In Figure 8, the performance of FPS, FetchSGD and BLCD is plotted for different data partitioning strategies under noise-free channel conditions on KDD10 dataset. Across all data partitioning scenarios it can been seen that BLCD and FPS perform equally well and better than FetchSGD.

Remarks on the results from KDD10 and KDD 12 datasets. The primary motivation to pick KDD10 and KDD12 datasets is to tackle the problem of feature selection in machine learning. Feature selection has numerous applications in a wide range of areas, including natural-language processing, genomics, and chemistry. In

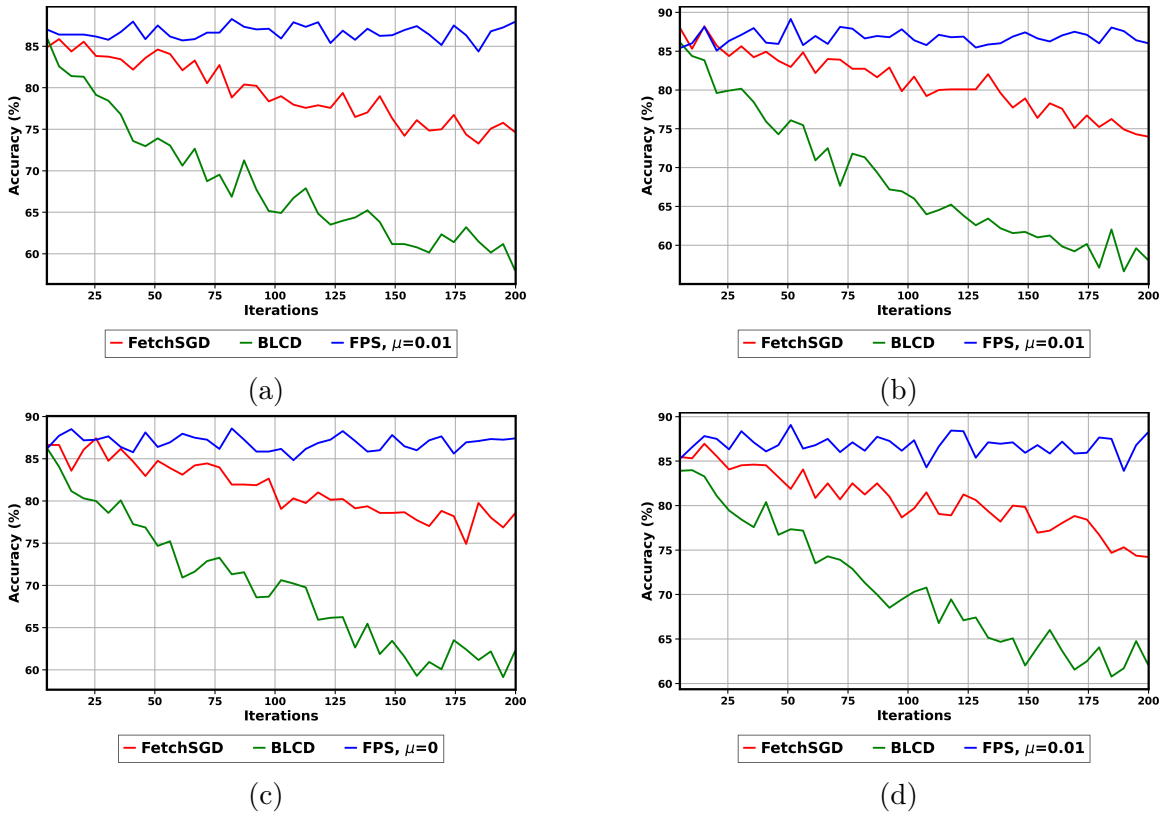


Figure 7. Plotting test accuracy for FPS, BLCD, FetchSGD on KDD10 dataset under noisy channel conditions. The figures correspond to different data partitioning strategies: (a) Scenario 1 (b) Scenario 2 (c) Scenario 3 (d) Scenario 4. I can see that FPS is stable under noisy channel conditions and consistently performs better than other competing bandlimited algorithms.

feature selection the goal is to identify a small subset of features that best models the relationship between the input data and output. Therefore, learning a small subset of features when the problem at hand in a higher dimension, requires efficient compression techniques. As a consequence, the gradient update vectors satisfy the approximately sparse assumption defined in Assumption 4.

For the real-world datasets considered in this paper (KDD10 and KDD12), It can be shown that the computed stochastic gradient vector at each iteration satisfies the approximately sparse gradient assumption (Assumption 4) in Appendix 2.7.2.

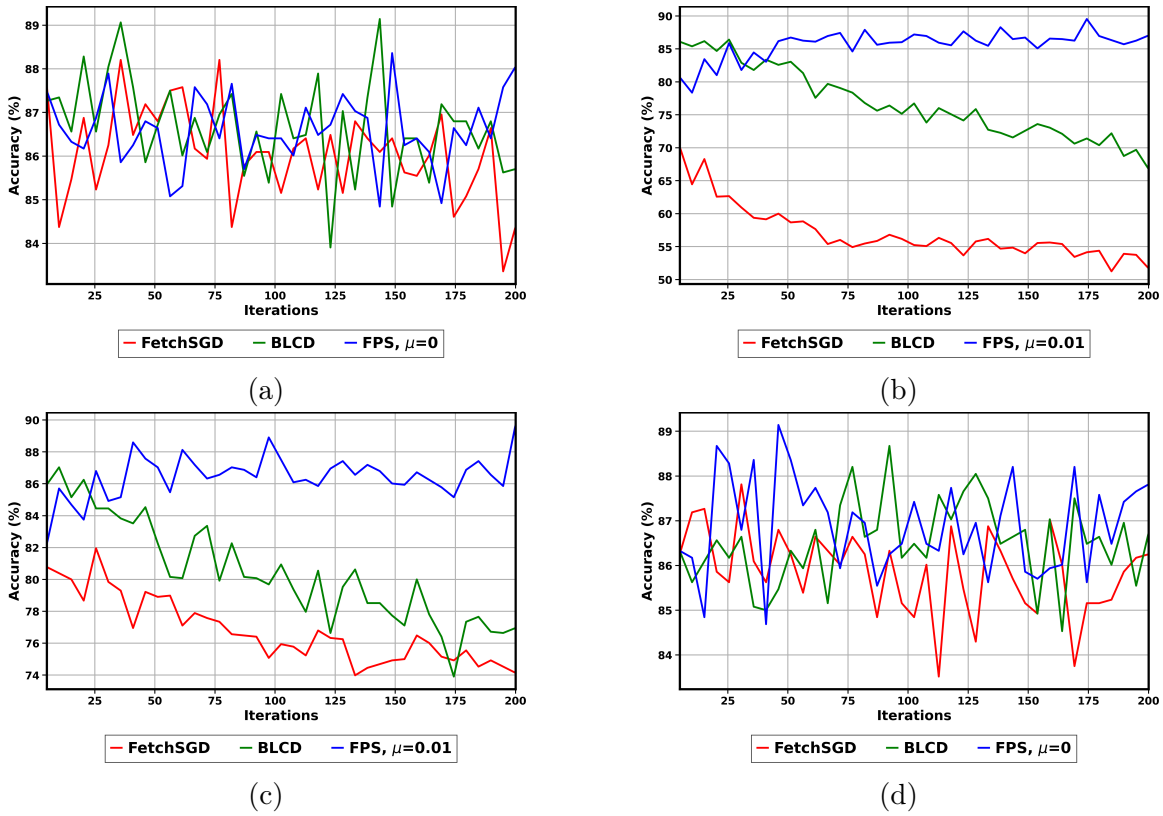


Figure 8. Plotting test accuracy for FPS, BLCD, FetchSGD on KDD10 dataset under noise-free channel conditions. The figures correspond to different data partitioning strategies: (a) Scenario 1 (b) Scenario 2 (c) Scenario 3 (d) Scenario 4.

Specifically for KDD12, the number of significant coordinates in the gradient update vectors are extremely low compared to the ambient dimension of the dataset. In this case, algorithms like BLCD will perform poorly as the probability of randomly selecting significant coordinates when the ambient dimension is huge, is very low. This poor performance of BLCD can be be seen in Figures 5 and 7. FetchSGD on the other hand maintains an efficient representation of significant coordinates of the gradient update vectors, so one would expect it to perform well. On the contrary, as FetchSGD contains no mechanism to tackle noisy wireless channels and data heterogeneity; it's performance is poor as well. The only scenarios where FetchSGD performs comparable

to our algorithm FPS, is when the data is distributed in an IID manner (Scenario 1) and the degree of statistical heterogeneity is low (Scenario 4), and the communication is over noise free channels (see Tables 1, 2). Accuracy plots of FetchSGD, BLCD and FPS under noise free case over varying degrees of data heterogeneity are shown in Figures 8 and 6.

The comparison can be taken a step further by evaluating FPS against FedProx T. Li et al. 2020 and top-k federated algorithms which are not bandlimited in nature. FedProx is one of the state-of-the-art algorithms recently published which aims to learn a global model when data heterogeneity exists across edge devices. FedProx communicates the whole gradient update vector with the central server and top-k algorithm requires extra rounds of communication between other edge devices to achieve consensus on global top-k gradient coordinates. The detailed accuracy results are given in Tables 1 and 2. When the data is extremely heterogeneous (Scenario 2), we see that FedProx and top-k do not perform well under both noisy and noise-free channel conditions. Under mild statistical heterogeneity setting like Scenario 4, we see that FedProx and top-k perform on par with our FPS algorithm in a noise-free channel setting, however, they struggle in noisy channel conditions. The hypothesis of poor performance of FedProx in noisy channel conditions due to approximately sparse gradient update vectors being corrupted by the channel noise. As the less significant coordinates are corrupted, this results in erroneous gradient updates. One can argue that this can be resolved by scaling the gradient coordinates well above the noise floor but this approach seems to be infeasible when there are power constraints imposed.

Label skewness	Noise $\mathcal{N}(0,\sigma^2)$	Accuracy (%)				
		FPS	FetchSGD	BLCD	Top-k	FedProx
Scenario 1	$\sigma=0$	96.44 ± 0.81	96.48 ± 1.52	8.51 ± 2.67	96.64 ± 0.52	96.48 ± 0.81
	$\sigma=1$	96.56 ± 1.29	5.46 ± 1.33	16.17 ± 21.23	68.82 ± 16.66	57.42 ± 13
Scenario 2	$\sigma=0$	97.03 ± 1.14	48.12 ± 1.26	5.93 ± 1.6	51.09 ± 2.93	53.20 ± 6.99
	$\sigma=1$	96.87 ± 0.95	5.39 ± 0.96	5.93 ± 1.85	57.57 ± 24.04	40.93 ± 11.12
Scenario 3	$\sigma=0$	96.64 ± 0.52	96.79 ± 0.51	6.56 ± 1.38	96.64 ± 1.22	96.56 ± 0.67
	$\sigma=1$	97.5 ± 0.97	5.39 ± 1.24	6.4 ± 1.27	72.57 ± 15.3	54.60 ± 17.26
Scenario 4	$\sigma=0$	96.25 ± 0.76	96.17 ± 1.08	17.18 ± 19.55	96.71 ± 0.46	96.01 ± 1.22
	$\sigma=1$	96.87 ± 0.95	6.09 ± 0.31	6.79 ± 1.93	66.32 ± 14.71	46.32 ± 10.79

Table 1. Test accuracy of different distributed algorithms under varying channel conditions and statistical heterogeneity. For FPS and FedProx, I tune μ from $\{0, 0.01, 0.1, 1\}$ and report the best accuracy over KDD 12 dataset.

Label skewness	Noise $\mathcal{N}(0,\sigma^2)$	Accuracy (%)				
		FPS	FetchSGD	BLCD	Top-k	FedProx
Scenario 1	$\sigma=0$	88.04 ± 1.53	86.64 ± 1.19	86.79 ± 2.45	87.10 ± 1.54	88.12 ± 2.35
	$\sigma=1$	87.96 ± 1.36	75.78 ± 3.84	63.20 ± 4.15	55.85 ± 6.15	55.46 ± 1.69
Scenario 2	$\sigma=0$	87.03 ± 1.66	54.37 ± 2.6	72.18 ± 4.02	54.06 ± 3.64	55 ± 1.73
	$\sigma=1$	88.12 ± 1.75	76.25 ± 3.18	62.03 ± 2.81	50.07 ± 3.089	56.71 ± 3.39
Scenario 3	$\sigma=0$	89.68 ± 1.75	75.54 ± 1.68	77.65 ± 3.21	78.35 ± 3.11	80.46 ± 2.26
	$\sigma=1$	87.42 ± 2.05	79.76 ± 3.40	62.42 ± 3.37	52.03 ± 6.01	54.14 ± 3.86
Scenario 4	$\sigma=0$	87.81 ± 1.96	86.25 ± 1.44	86.95 ± 1.72	88.28 ± 1.71	88.43 ± 1.12
	$\sigma=1$	88.28 ± 2.06	76.71 ± 7.15	64.76 ± 2.11	59.37 ± 5.78	56.32 ± 3.6

Table 2. Test accuracy of different distributed algorithms under varying channel conditions and statistical heterogeneity. For FPS and FedProx, I tune μ from $\{0, 0.01, 0.1, 1\}$ and report the best accuracy over KDD 10 dataset.

2.6.2.3 MNIST Dataset

Now let us consider a widely used dataset in machine learning literature –MNIST– and observe the performance of our algorithm on it. For this a simple 2-layer neural network is designed with number of parameters (neurons) $\sim 100,000$. For communication efficient algorithms (FPS, FetchSGD, BLCD) the number of subcarriers are vary from $\{5000, 10000, 20000\}$. The regularization parameter (μ) for proximal term takes the values : $\{0, 0.01, 0.1, 1\}$. For count-sketch algorithms (FPS, FetchSGD) the number of top$-k$ heavy hitters extracted varies from : $\{2000, 5000, 10000\}$. In Figure 10, the performance averaged over 3 trials of different band-limited algorithms including

FPS under noisy wireless channel case over varying degrees of data heterogeneity in plotted. An in detail comparison is provided in Table 3 where the other competing FL algorithms are considered as well. In Figure 9, the performance of FPS, FetchSGD and BLCD for different data heterogenity scenarios mentioned earlier under noise-free channel conditions on MNIST dataset in plotted. Across all scenarios it can be seen that FPS performs well against its band-limited competitors FetchSGD and BLCD.

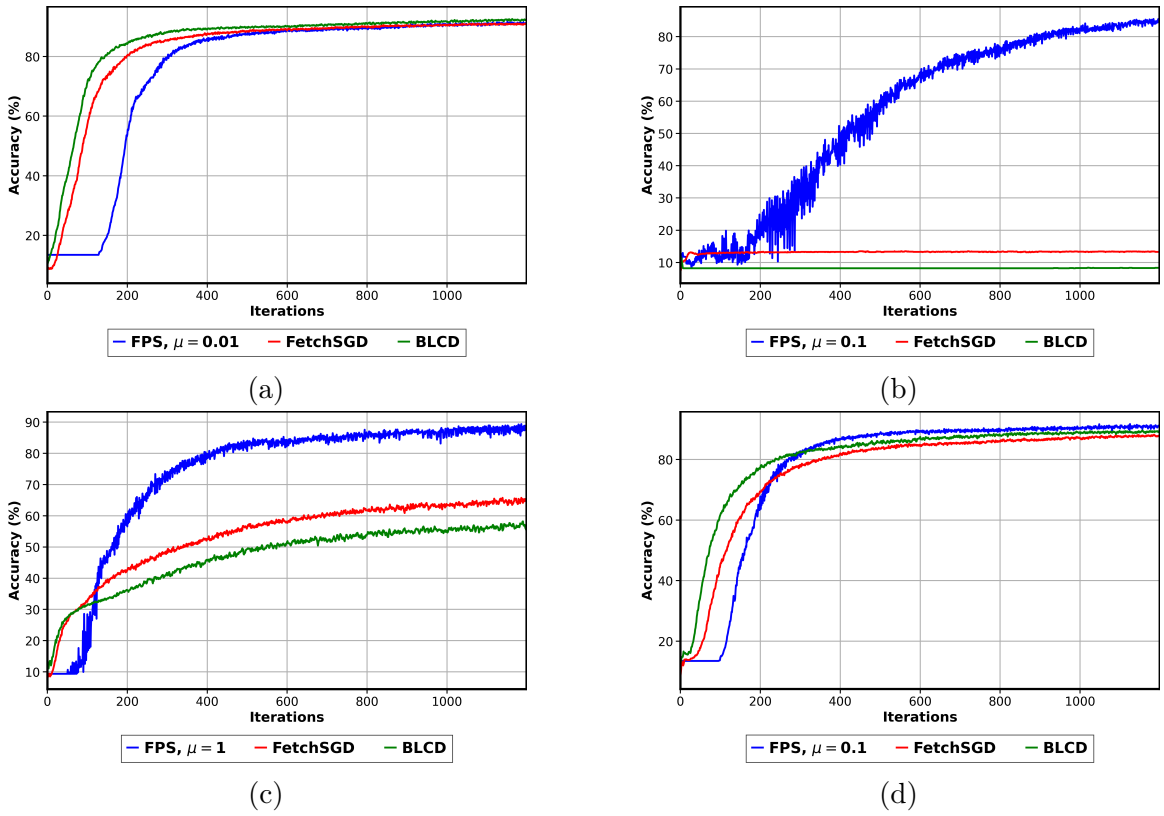


Figure 9. Plotting test accuracy for FPS, BLCD, FetchSGD on MNIST dataset under noise-free channel conditions. The figures correspond to different data partitioning strategies: (a) Scenario 1 (b) Scenario 2 (c) Scenario 3 (d) Scenario 4.

In the noisy IID case, as depicted in Figure 10a, both FPS and FetchSGD perform well and have comparable accuracies. The slower convergence of FPS can be attributed to the sketching of gradient updates into the count sketch operator, which leads to

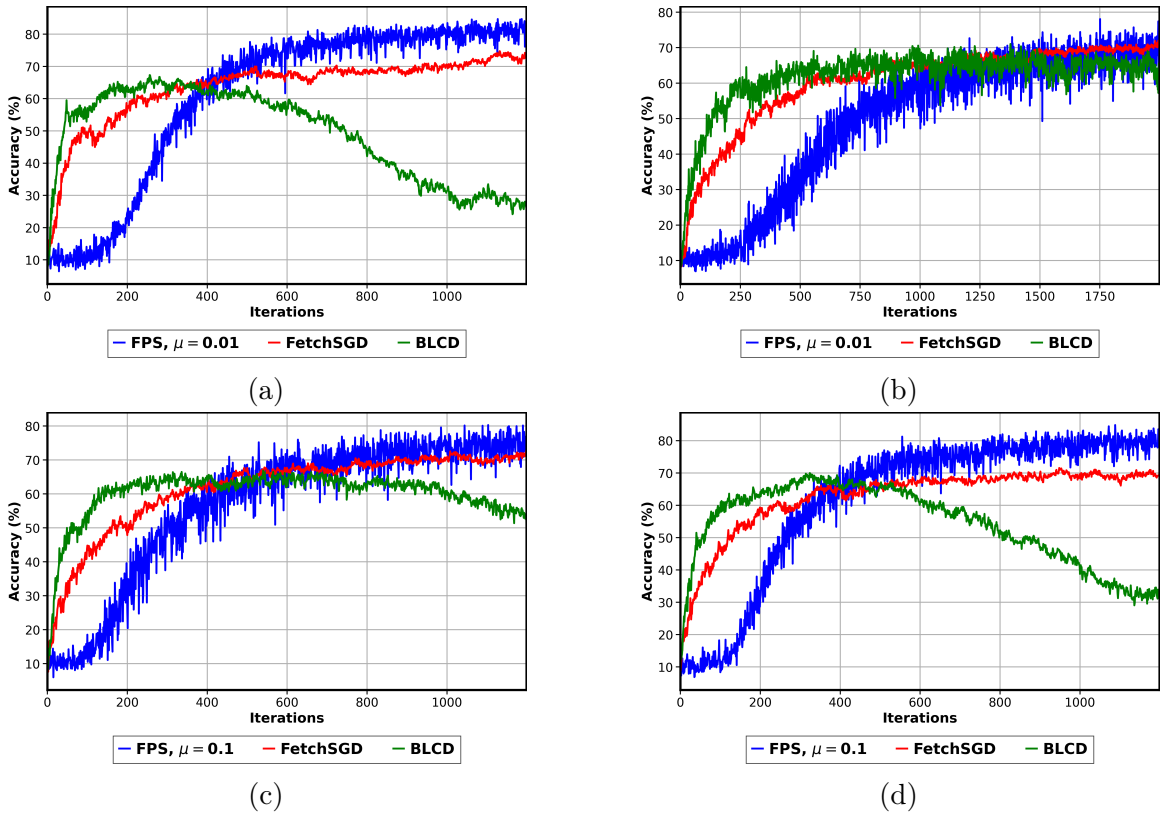


Figure 10. Plotting test accuracy for FPS, BLCD, FetchSGD on MNIST dataset under noisy channel conditions. The figures correspond to different data partitioning strategies: (a) Scenario 1 (b) Scenario 2 (c) Scenario 3 (d) Scenario 4. It can be seen that FPS is stable under noisy channel conditions and consistently performs better than or on-par with other competing bandlimited algorithms.

cancellations and an efficient representation of model parameters at each epoch. Therefore, by leveraging the existence of a low-dimensional representation of model parameters and consistently sketching gradient vectors in the count-sketch data structure (without re-initializing the CS data structure to zero at each communication round's start), the low-dimensional model parameters can be efficiently represented after a few epochs. Also, observe that the BLCD algorithm exhibits a sudden increase in accuracy followed by a rapid decline in performance. The hypothesis is that BLCD can quickly learn optimal model parameters, resulting in minimal gradient updates.

However, it is expected that noisy wireless channels may corrupt these updates, potentially causing model divergence or a drop in accuracy.

In scenario 2, for the MNIST dataset, comprising 10 classes distributed across 10 edge devices in such a way that each edge device has samples corresponding to only one class, as illustrated in Figure 10b, it can be observed that the proposed algorithm FPS maintains its robust performance in the presence of extreme data heterogeneity. The intriguing aspect is that even though BLCD and FetchSGD are not equipped to handle data heterogeneity amongst clients, they perform well. Analyzing this behavior can be an interesting open question and is left for future work.

In Scenarios 3 and 4, where data heterogeneity is not extreme, Figures 10c and 10d illustrate that FPS performs better or comparably to FetchSGD. BLCD does not perform well, exhibiting behavior similar to that observed in the noisy IID case.

Table 3 indicates that, overall, FPS is more robust and consistently performs well across different cases. Notably, algorithms like FedProx and Top-k, which are not inherently band-limited, show effective performance only in the IID setting or Scenario 4, where the level of data heterogeneity is very mild. Despite the close relationship with the FedProx algorithm in terms of employing the proximal term, the results suggest that count-sketch data structures are robust to additive noise in wireless channels and provide reliable estimates of model parameters.

2.7 Discussion and Future Extensions

2.7.1 Choosing Hyperparameters

There are two hyperparameters that we consider in the main paper that require further discussion. The first one is the choice of proximal parameter, μ. A large value of μ will cause the future iterates to be close to the initialization iterate and a low value of

Label skewness	Noise $\mathcal{N}(0, \sigma^2)$	Accuracy (%)				
		FPS	FetchSGD	BLCD	Top-k	FedProx
Scenario 1	$\sigma = 0$	90.23 ± 1.79	91.01 ± 3.01	92.38 ± 0.57	97.33 ± 4.8	91.53 ± 1.26
	$\sigma = 0.8$	81.31 ± 1.3	74.21 ± 0.87	28.05 ± 1.29	13.80 ± 5.61	66.47 ± 3.22
Scenario 2	$\sigma = 0$	84.5 ± 0.82	13.2 ± 1.33	8.33 ± 1.61	10.48 ± 3.82	8.2 ± 1.41
	$\sigma = 0.8$	74.54 ± 1.52	71.80 ± 1.79	66.40 ± 1.26	63.54 ± 4.76	8.39 ± 0.36
Scenario 3	$\sigma = 0$	87.63 ± 0.57	65.69 ± 0.97	56.83 ± 1.52	64.97 ± 3.91	28.38 ± 0.48
	$\sigma = 0.8$	78.38 ± 0.64	72.72 ± 3.53	54.03 ± 1.11	15.8 ± 6.09	43.16 ± 1.7
Scenario 4	$\sigma = 0$	90.36 ± 0.54	88.02 ± 2.5	89.38 ± 2.5	95.7 ± 2.76	89.84 ± 0.15
	$\sigma = 0.8$	80.27 ± 0.80	70.83 ± 2.48	33.9 ± 1.7	11.71 ± 7.3	67.57 ± 0.9

Table 3. Test accuracy of different distributed algorithms under varying channel conditions and statistical heterogeneity. For FPS and FedProx, the proximal parameter μ is chosen from $\{0, 0.01, 0.1, 1\}$ and report the best accuracy over MNIST dataset.

μ may cause the model to diverge. Therefore, the value of proximal parameter must be chosen carefully. In our experiments, we choose the best value of this proximal parameter from a set of values {0, 0.01, 0.1, 1}. For the two real-world data sets (KDD10 and KDD12) across different data partitioning strategies the best values of μ are 0.01 and 1 respectively. Note that picking the best value of μ right away is difficult due to varying statistical heterogeneity and different datasets. An interesting line of work could be finding the ideal choice of proximal parameter automatically. However, another interesting heuristic technique proposed in T. Li et al. 2020 adaptively tunes μ. For instance, increase μ when the loss increases and vice versa. We have not examined the effects of such a heuristic in our experiments.

Another hyperparameter that we choose prior to the start of our experiments is number of local updates E performed by each edge device. We choose a uniform $E = 5$ across all edge devices. Choosing a large value of E implies allowing large amounts of work done by edge devices and this can cause the model to diverge when the data is distributed in a non-IID manner. However, to mitigate this we have a proximal term which does not allow the local updates performed by the edge devices in this period to drift far away. However, the choice of an appropriate value of E

might be challenging problem in itself as it depends on device constraints and data distribution across all devices.

2.7.2 Gradient Compressibility

The idea that the computed stochastic gradients are compressible or approximately sparse is central to employ efficient compression techniques. In the main paper we formulate mathematically the approximately sparse behaviour of the computed gradients. This needs to be empirically validated as well. We consider the scenario where the data is distributed in an IID manner across devices. We run a federated learning algorithm where there is no bandwidth limitation i.e., high-dimensional gradient vectors are communicated. We consider noise-free channels and the updates are communicated to the central server at every iteration. The loss function has no proximal term appended to it. This naive setup will help us understand the true behaviour of computed stochastic gradients. We run this vanilla FL algorithm for 200 iterations and at the end of it we report $\sim 90\%$ accuracy on both real world datasets (KDD10 and KDD12).

The number of features in the datasets KDD10 and KDD12 are 20,216,830 and 54,686,452 respectively. In Figures 11(a) and 12(a), we plot the absolute value of gradient coordinates computed at a particular edge device for the datasets KDD10 and KDD12 respectively. This plot is captured across three time instants, at iteration 25, 75 and 150. We see that in both figures, the absolute value of coordinates of the local gradient vector sorted in decreasing order are approximately sparse or follow a power law distribution. Similarly in Figures 11(b) and 12(b) we plot the absolute value of coordinates of the aggregated gradient vector received at the central server sorted in decreasing order. This plot is captured across three time instants, at iteration 25,

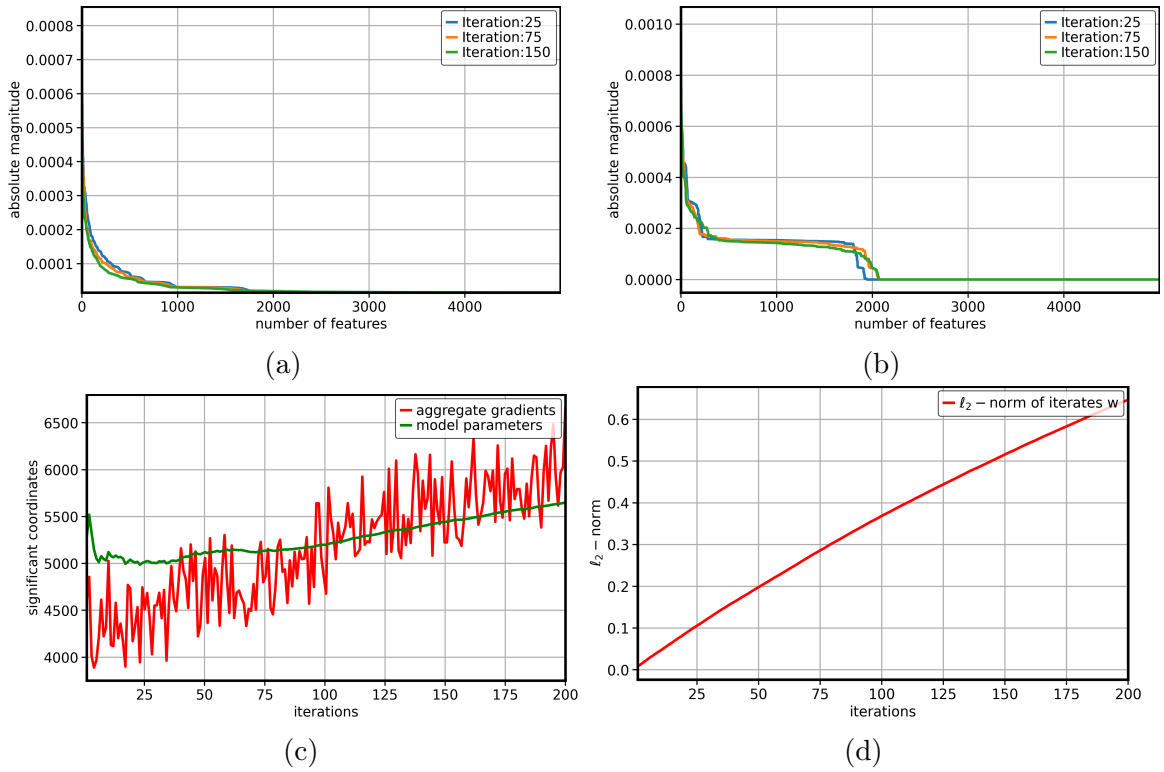


Figure 11. KDD 10 Dataset (a) sorted stochastic gradient at a single edge device (b) sorted aggregated stochastic gradient at the central server (c) significant coordinates of aggregated gradient vector and iterates at the central server (d) ℓ_2- norm of iterates.

75 and 150. We observe a similar approximately sparse or power law behaviour for aggregated gradient vectors. If we approximate the number of significant coordinates in computed gradient vectors just by visual inspection of the plots, it is less than 3000. This is far less than the ambient dimension of the datasets we are operating on.

However, a stronger notion of significant coordinates needs to be used. To this extent we use an alternative measure called *soft* sparsity defined in Lopes 2016:

$$sp(\mathbf{x}) = \frac{\|\mathbf{x}\|_1^2}{\|\mathbf{x}\|_2^2} \tag{2.15}$$

Soft-sparsity represents the number of significant coordinates in a vector. Let $\mathbf{g}$ and $\mathbf{w}$ denote the aggregated gradient and the model parameter vector respectively. For KDD10 dataset, the number of significant coordinates for the aggregated gradient

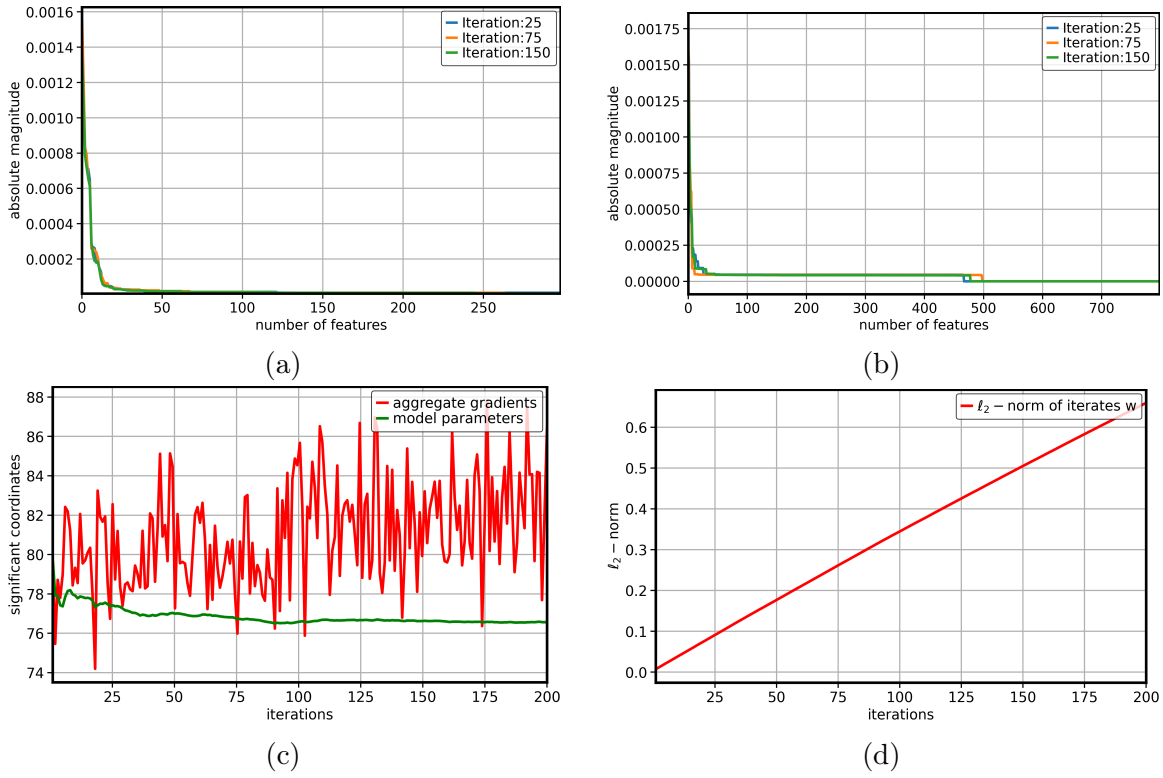


Figure 12. KDD 12 Dataset (a) sorted stochastic gradient at a single edge device (b) sorted aggregated stochastic gradient at the central server (c) significant coordinates of aggregated gradient vector and iterates at the central server (d) ℓ_2- norm of iterates.

vector $sp(\mathbf{g})$ and the model parameter vector $sp(\mathbf{w})$ are $\sim$ 5000, which is much smaller than the ambient dimension. Similarly, for KDD12 dataset, the the number of significant coordinates for the aggregated gradient vector $sp(\mathbf{g})$ are ~ 85 and the model parameter vector $sp(\mathbf{w})$ are ~ 75. This can be seen in Figures 11(c) and 12(c).

Additionally, we show that the ℓ_2-norm of the iterates at every iteration received at the central server does not explode and can be uniformly bounded above by a constant. This can be seen in Figures 11(d) and 12(d) for datasets KDD10 and KDD12 respectively.

2.7.3 Limitations of Federated Proximal Sketching

It is also essential to highlight the limitations of the proposed approach. One of the pivotal assumptions for our algorithm to work is the approximately sparse gradient vector assumption and that the model parameters lie in a low-dimensional space. This assumption minimizes the number of collisions between heavy coordinates in the count sketch data structure while sketching. Therefore, in applications where gradient vectors are dense, a compression technique like count sketch will not work and FL algorithms like top$-k$ and BLCD will outperform. It is important to note that each of the state-of-the-art algorithms mentioned above perform well in specific scenarios discussed. Based on the application setting, it is the decision of the user to select a FL algorithm that facilitates learning.

2.8 Conclusion

In this chapter, the proposed method, Federated Proximal Sketching (FPS), is a novel algorithm designed to effectively learn a global model while accounting for the challenges posed by bandlimited noisy wireless channels and data heterogeneity present across edge devices. This research provides both theoretical guarantees and empirical results, establishing a foundation for using sketching as a compression operator in scenarios characterized by limited communication bandwidth and heterogeneous data distributions.

The theoretical analysis demonstrates that the communication cost to the central server during each iteration is $\mathcal{O}(\log d)$. This cost is significantly lower than the ambient dimension d, making FPS particularly advantageous for large-scale datasets where efficient communication is crucial. Empirical results further corroborate the effectiveness of the count-sketch compression scheme integrated into FPS, highlight-

ing its ability to significantly reduce communication costs without any noticeable degradation in model performance.

To simulate data heterogeneity across edge devices, various data partitioning strategies motivated by real-world scenarios are considered. The restructuring of the loss function through the incorporation of a proximal term plays a critical role in stabilizing the algorithm, preventing divergence under varying degrees of data heterogeneity and the influence of channel noise. Mathematically, the effects of data heterogeneity and the bias induced by channel noise are modeled using mild technical assumptions, resulting in an easily interpretable convergence result. This result elucidates the interplay among several key parameters, including the size of the count sketch data structure, the degree of statistical heterogeneity, the magnitude of the induced bias, and the rate of convergence.

Overall, this work adeptly tackles three of the most pressing challenges in federated learning setup: data heterogeneity across edge devices, bandlimited and noisy wireless channels, and demonstrates the robustness and efficacy of our proposed algorithm - FPS. Our experiments conducted over synthetic and large-scale real-world datasets, substantiate our theoretical guarantees and showcase the superior, stable and highly accurate performance of FPS over other state-of-the-art federated learning algorithms.

2.9 Proofs of Theorems and Lemmas

This section provides detailed proofs of theorems and lemmas discussed in Chapter 2

2.9.1 Count Sketch Recovery Guarantees

The main theorem of Charikar et al. 2002 is stated here which highlights the recovery guarantees of count sketch data structure.

Theorem 3 (Count sketch) *For a vector $\mathbf{g} \in \mathbb{R}^d$, count sketch recovers the top-k coordinates with error $\pm\varepsilon\|\mathbf{g}\|_2$ with memory $\mathcal{O}\left(\left(k + \frac{\|\mathbf{g}^{tail}\|^2}{\varepsilon^2 \mathbf{g}(k)^2}\right) \log \frac{d}{\delta}\right)$; where $\|\mathbf{g}^{tail}\|^2 = \sum_{i \notin top-k}(\mathbf{g}(i))^2$ and $\mathbf{g}(k)$ is the k-th largest coordinate and this holds with probability at least $1 - \delta$.*

Proof:

Setup:

1. Let $h_1, ..., h_r : [d] \to [b]$ be pairwise independent hash functions.
2. Let $s_1, ..., s_r : [d] \to \{-1, +1\}$ be pairwise independent sign functions.
3. The Count-Sketch data structure is an $r \times b$ array of counters $C[i, j]$.

Estimation Procedure: For each coordinate i, we estimate $\mathbf{g}(i)$ as:

$$\hat{\mathbf{g}}(i) = \text{median}_j\{C[j, h_j(i)] \cdot s_j(i)\}$$

Error Analysis: Let's analyze the error for a fixed coordinate i.

1. For each j, define $X_j = C[j, h_j(i)] \cdot s_j(i)$.
2. We can express X_j as:

$$X_j = \mathbf{g}(i) + \sum_{k \neq i} \mathbf{g}(k) \cdot s_j(k) \cdot \mathbb{I}[h_j(k) = h_j(i)]$$

3. The expectation of X_j is $g(i)$:

$$\mathbb{E}[X_j] = \mathbf{g}(i) + \sum_{k \neq i} \mathbf{g}(k) \cdot \mathbb{E}[s_j(k) \cdot \mathbb{I}[h_j(k) = h_j(i)]] = g(i)$$

4. The variance of X_j can be bounded:

$$\text{Var}(X_j) \leq \frac{1}{b} \sum_{k \neq i} \mathbf{g}(k)^2 \leq \frac{\|\mathbf{g}^{tail}\|^2}{b}$$

Concentration Bound: By setting $b = O(\|\mathbf{g}^{tail}\|^2/(\varepsilon^2\mathbf{g}(k)^2))$, we ensure that the variance is small enough.

Using Chebyshev's inequality:

$$\mathbb{P}(|X_j - \mathbf{g}(i)| > \varepsilon\|\mathbf{g}\|_2) \leq \frac{\mathrm{Var}(X_j)}{\varepsilon^2\|\mathbf{g}\|_2^2} \leq \frac{1}{4}$$

By taking the median of $r = O(\log(dT/\delta))$ independent repetitions, we can boost this probability:

$$\mathbb{P}(|\hat{\mathbf{g}}(i) - \mathbf{g}(i)| > \varepsilon\|\mathbf{g}\|_2) \leq \delta/d$$

Recovery Guarantee: Applying the union bound over all d coordinates:

$$\mathbb{P}(\exists i : |\hat{\mathbf{g}}(i) - \mathbf{g}(i)| > \varepsilon\|\mathbf{g}\|_2) \leq \delta$$

This implies that with probability at least $1 - \delta$, for all coordinates i:

$$|\hat{\mathbf{g}}(i) - \mathbf{g}(i)| \leq \varepsilon\|\mathbf{g}\|_2$$

Space Complexity: The total space used is:

$$\mathcal{O}(rb) = \mathcal{O}\left(\left(k + \frac{\|\mathbf{g}^{tail}\|^2}{\varepsilon^2\mathbf{g}(k)^2}\right)\log\frac{d}{\delta}\right)$$

This completes the proof of the Count-Sketch theorem, showing both the error guarantee and the space complexity. ■

2.9.2 Residual Unsketching Error (Proof of Lemma 1)

The following lemma provides an upper bound on the residual error after unsketching the top-k coordinates of the iterates. This result directly follows from the initial recovery guarantees established in Theorem 1. Let the expected value of iterates at any time $t \geq 0$ be uniformly bounded above by a positive constant W. Formally, this can be stated as an assumption:

Assumption 6 (Bounded iterates) *Let $\mathbf{w}_t$ denote the weight vector at time t (or the t-th iterate). The assumption states that the expected squared norm of $\mathbf{w}_t$ is bounded by a constant W, uniformly over all t:*

$$\mathbb{E}\left[\|\mathbf{w}_t\|_2^2\right] \leq W, \quad \forall t \in \mathbb{N},$$

where, W is a positive constant and $\mathbb{E}$ denotes the expectation with respect to any randomness in $\mathbf{w}_t$ (e.g., from stochastic updates or random initialization).

While this assumption may appear strong, the inclusion of a regularizer and the use of the count-sketch data structure to compress model parameters in the proposed approach support its reasonableness. Moreover, empirical validation of bounded iterates in Section 2.7.2 further confirms its validity. Denote by $\tilde{\mathbf{w}}_t$ the unsketched top-k coordinates of the iterate $\mathbf{w}_t$, where t represents the time index. Under Assumption 4 and the count-sketch recovery guarantees from Theorem 1, the following lemma holds:

Lemma 1 *For any $c > 0$, if the count sketch data structure recovers the top-k coordinates with error $\varepsilon = \frac{1}{\sqrt{c\,k}}$ and sketch size scaling like $\mathcal{O}\left(c\,k\,\log\frac{dT}{\delta}\right)$, the following holds for iterate $\mathbf{w}_t \in \mathbb{R}^d$, $\forall t \geq 1$ with probability at least $1-\delta$:*

$$\mathbb{E}\left[\|\mathbf{w}_t - \tilde{\mathbf{w}}_t\|^2\right] \leq \left(\frac{1}{c} + \frac{(k+1)^{1-2p} - d^{1-2p}}{2p-1}\right) W, \tag{2.16}$$

where, $\tilde{\mathbf{w}}_t = \mathcal{U}(S(\mathbf{w}_t))$ is top-k unsketched vector.

Proof:

$$
\begin{aligned}
\mathbb{E}\left[\|\mathbf{w}_t - \tilde{\mathbf{w}}_t\|_2^2\right] &= \mathbb{E}\left[\|\mathbf{w}_t - \mathcal{U}_k(S(\mathbf{w}_t))\|^2\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{k} \mid \mathbf{w}_t(i) - \tilde{\mathbf{w}}_t(i) \mid^2 + \sum_{i=k+1}^{d} (\mathbf{w}_t(i))^2\right] \\
&= \mathbb{E}\left[\varepsilon^2\, k\, \|\mathbf{w}_t\|^2 + \sum_{i=k+1}^{d} (\mathbf{w}_t(i))^2\right] \\
&= \mathbb{E}\left[\varepsilon^2\, k\, \|\mathbf{w}_t\|^2 + \sum_{i=k+1}^{d} i^{-2p} \left(\sum_{j=1}^{t} ||-\gamma\, \mathbf{g}_j||\right)^2\right] \\
&\leq \mathbb{E}\left[\varepsilon^2\, k\, \|\mathbf{w}_t\|^2 + \sum_{i=k+1}^{d} i^{-2p} \left(\left|\left|\sum_{j=1}^{t} -\gamma\, \mathbf{g}_j\right|\right|\right)^2\right] \\
&= \left(\varepsilon^2 k + \sum_{i=k+1}^{d} i^{-2p}\right) \mathbb{E}\left[\|\mathbf{w}_t\|^2\right] \\
&\leq \left(\frac{1}{c} + \frac{(k+1)^{1-2p} - d^{1-2p}}{2p-1}\right) \mathbb{E}\left[\|\mathbf{w}_t\|^2\right] \\
&\leq \left(\frac{1}{c} + \frac{(k+1)^{1-2p} - d^{1-2p}}{2p-1}\right) W\,. \qquad (2.17)
\end{aligned}
$$

■

As the sketch size increases, more coordinates can be recovered with higher accuracy, which leads to a reduction in the residual error ε. A larger sketch size effectively captures a more detailed representation of the original data, thus enabling a more precise unsketching process. Additionally, as the gradients become more compressible—characterized by an increase in the compressibility parameter p—the residual error decreases further. This is because higher compressibility implies that the essential information in the gradients can be captured in fewer coordinates, improving the accuracy of the reconstruction from the sketch.

2.9.3 Drift in Iterates during Local Training (Proof of Lemma 2)

Lemma 2 *For a step size* $\gamma \leq \frac{1}{4\,E\,(L+\mu)\,(1+2\,B^2(P_b+P_n))}$ *, the drift in iterates for any* $e \in \{0, \ldots, E-1\}$ *can be bounded above as,*

$$\mathbb{E}\left[\|\mathbf{w}_t^e - \mathbf{w}_t\|^2\right] \leq 30\,E^2\,\gamma^2\,\left((1+2\,B^2\,(P_b+P_n))\|\nabla f(\mathbf{w}_t)\|^2 + b^2 + \sigma^2\right) \tag{2.18}$$

Proof: We focus on the term $\|\mathbf{w}_t^e - \mathbf{w}_t\|^2$, I get:

$$\begin{aligned}
&\mathbb{E}\left[\|\mathbf{w}_t^e - \mathbf{w}_t\|^2\right] \\
&= \mathbb{E}\left[\|\mathbf{w}_t^{e-1} - \gamma\,\mathbf{g}_t^{e-1} - \mathbf{w}_t\|^2\right] \\
&= \mathbb{E}\left[\|\mathbf{w}_t^{e-1} - \mathbf{w}_t - \gamma\,\left(\mathbf{g}_t^{e-1} - \nabla f(\mathbf{w}_t^{e-1}) + \nabla f(\mathbf{w}_t^{e-1}) - \nabla f(\mathbf{w}_t) + \nabla f(\mathbf{w}_t)\right)\|^2\right] \\
&\leq \left(1 + \frac{1}{2E-1}\right)\,\mathbb{E}\left[\|\mathbf{w}_t^{e-1} - \mathbf{w}_t\|^2\right] \\
&\quad + 2\,E\,\gamma^2\,\mathbb{E}\left[\|\nabla f(\mathbf{w}_t^{e-1}) - \mathbf{g}_t^{e-1} + \nabla f(\mathbf{w}_t^{e-1}) - \nabla f(\mathbf{w}_t) + \nabla f(\mathbf{w}_t)\|^2\right] \\
&\leq \left(1 + \frac{1}{2E-1}\right)\,\mathbb{E}\left[\|\mathbf{w}_t^{e-1} - \mathbf{w}_t\|^2\right] + 6\,E\,\gamma^2\,\mathbb{E}\left[\|\nabla f(\mathbf{w}_t^{e-1}) - \mathbf{g}_t^{e-1}\|^2\right] \\
&\quad + 6\,E\,\gamma^2\,\mathbb{E}\left[\|\nabla f(\mathbf{w}_t^{e-1}) - \nabla f(\mathbf{w}_t)\|^2\right] + 6\,E\,\gamma^2\,\|\nabla f(\mathbf{w}_t)\|^2 \\
&\leq \left(1 + \frac{1}{2E-1} + 6\,E\,(L+\mu)^2\,\gamma^2\right)\,\mathbb{E}\left[\|\mathbf{w}_t^{e-1} - \mathbf{w}_t\|^2\right] \\
&\quad + 6\,E\,\gamma^2\,\left(\mathbb{E}\left[\|\beta_t^{e-1} + \zeta_t^{e-1}\|^2\right] + \|\nabla f(\mathbf{w}_t)\|^2\right) \\
&\leq \left(1 + \frac{1}{2E-1} + 6\,E\,(L+\mu)^2\,\gamma^2\right)\,\mathbb{E}\left[\|\mathbf{w}_t^{e-1} - \mathbf{w}_t\|^2\right] \\
&\quad + 6\,E\,\gamma^2\,\left(B^2\,(P_b+P_n)\,\mathbb{E}\left[\|\nabla f(\mathbf{w}_t^{e-1})\|^2\right] + b^2 + \sigma^2 + \|\nabla f(\mathbf{w}_t)\|^2\right) \\
&\leq \left(1 + \frac{1}{2E-1} + 6\,E\,(L+\mu)^2\,\gamma^2\right)\,\mathbb{E}\left[\|\mathbf{w}_t^{e-1} - \mathbf{w}_t\|^2\right] \\
&\quad + 6\,E\,\gamma^2\,\left(B^2\,(P_b+P_n)\,\mathbb{E}\left[\|\nabla f(\mathbf{w}_t^{e-1}) - \nabla f(\mathbf{w}_t) + \nabla f(\mathbf{w}_t)\|^2\right] + b^2 + \sigma^2 + \|\nabla f(\mathbf{w}_t)\|^2\right) \\
&\leq \left(1 + \frac{1}{2E-1} + 6\,E\,(L+\mu)^2\,\gamma^2\,(1+2\,B^2\,(P_b+P_n))\right)\,\mathbb{E}\left[\|\mathbf{w}_t^{e-1} - \mathbf{w}_t\|^2\right] \\
&\quad + 6\,E\,\gamma^2\,\left((1+2\,B^2\,(P_b+P_n))\|\nabla f(\mathbf{w}_t)\|^2 + b^2 + \sigma^2\right)\,.
\end{aligned}$$

Additionally, assume $\gamma \leq \frac{1}{4\,E\,(L+\mu)\,(1+2\,B^2(P_b+P_n))}$ and using this in the analysis we get:

$$\begin{aligned}
\mathbb{E}\,\left[\|\mathbf{w}_t^e - \mathbf{w}_t\|^2\right] &\leq \left(1 + \frac{1}{2E-1} + \frac{6}{16\,(1+2\,B^2\,(P_b+P_n))E}\right)\mathbb{E}\,\left[\|\mathbf{w}_t^{e-1} - \mathbf{w}_t\|^2\right] \\
&\quad + 6\,E\,\gamma^2\,\left((1+2\,B^2\,(P_b+P_n))\|\nabla f(\mathbf{w}_t)\|^2 + b^2 + \sigma^2\right) \\
&\leq \left(1 + \frac{1}{2E-1} + \frac{1}{2\,E}\right)\mathbb{E}\,\left[\|\mathbf{w}_t^{e-1} - \mathbf{w}_t\|^2\right] \\
&\quad + 6\,E\,\gamma^2\,\left((1+2\,B^2\,(P_b+P_n))\|\nabla f(\mathbf{w}_t)\|^2 + b^2 + \sigma^2\right) \\
&\leq \left(1 + \frac{1}{E-1}\right)\mathbb{E}\,\left[\|\mathbf{w}_t^{e-1} - \mathbf{w}_t\|^2\right] \\
&\quad + 6\,E\,\gamma^2\,\left((1+2\,B^2\,(P_b+P_n))\|\nabla f(\mathbf{w}_t)\|^2 + b^2 + \sigma^2\right)\,.
\end{aligned}$$

Going recursively,

$$\begin{aligned}
&\mathbb{E}\,\left[\|\mathbf{w}_t^e - \mathbf{w}_t\|^2\right] \\
&\leq \sum_{e=0}^{E-1}\left(1 + \frac{1}{E-1}\right)^e\,6\,E\,\gamma^2\,\left((1+2\,B^2\,(P_b+P_n))\|\nabla f(\mathbf{w}_t)\|^2 + b^2 + \sigma^2\right) \\
&= 6E(E-1)\left(\left(1 + \frac{1}{E-1}\right)^E - 1\right)\gamma^2\left((1+2B^2(P_b+P_n))\|\nabla f(\mathbf{w}_t)\|^2 + b^2 + \sigma^2\right) \\
&\leq 30\,E^2\,\gamma^2\,\left((1+2\,B^2\,(P_b+P_n))\|\nabla f(\mathbf{w}_t)\|^2 + b^2 + \sigma^2\right). \qquad (2.19)
\end{aligned}$$

The last inequality follows from the fact that $\left(1 + \frac{1}{E-1}\right)^E \leq 5$ for all $E > 1$. The proof of the above Lemma loosely follows the proof of Lemma 3 in Reddi et al. 2021. ■

2.9.4 Bounded Gradient Norm

Let us now bound the second moment bounds of computed stochastic gradient, bias and noise terms.

$$\mathbf{g}_t(\mathbf{w}_t) = \sum_{e=0}^{E-1}\mathbf{g}_t(\mathbf{w}_t^e)\,. \qquad (2.20)$$

Keeping the representation simple, we write $\mathbf{g}_t(\mathbf{w}_t^e) = \mathbf{g}_t^e$. Extending this representation, we can expand the computed gradient based on the general structure as, $\mathbf{g}_t^e = \nabla f(\mathbf{w}_t^e) + \beta_t^e + \zeta_t^e$.

$$\begin{aligned}
\mathbb{E}\left[\|\mathbf{g}_t\|^2\right] &= \mathbb{E}\left[\left\|\sum_{e=0}^{E-1} \nabla f(\mathbf{w}_t^e) + \beta_t^e + \zeta_t^e\right\|^2\right] \\
&\leq E\left(\sum_e \left((2 + 2P_b\,B^2 + P_n\,B^2)\|\nabla f(\mathbf{w}_t^e)\|^2 + 2b^2 + \sigma^2\right)\right) \\
&= (2 + 2P_b\,B^2 + P_n\,B^2)\,E\left(\sum_e \|\nabla f(\mathbf{w}_t^e)\|^2\right) + 2\,E^2\,b^2 + E^2\,\sigma^2 \\
&= (2 + 2P_b\,B^2 + P_n\,B^2)\,E\left(\sum_e \|\nabla f(\mathbf{w}_t^e) - \nabla f(\mathbf{w}_t) + \nabla f(\mathbf{w}_t)\|^2\right) \\
&\qquad + 2\,E^2\,b^2 + E^2\,\sigma^2 \\
&\leq 2\,(2 + 2P_b\,B^2 + P_n\,B^2)\,\underbrace{E\left(\sum_e \|\nabla f(\mathbf{w}_t^e) - \nabla f(\mathbf{w}_t)\|^2\right)}_{\text{term 1}} \\
&\qquad + 2\,(2 + 2P_b\,B^2 + P_n\,B^2)\,E^2\,\|\nabla f(\mathbf{w}_t)\|^2 + 2\,E^2\,b^2 + E^2\,\sigma^2\,.
\end{aligned} \tag{2.21}$$

Focusing on bounding term 1 in the above equation, we get:

$$E\left(\sum_e \mathbb{E}\left[\|\nabla f(\mathbf{w}_t^e) - \nabla f(\mathbf{w}_t)\|^2\right]\right) \leq (L + \mu)^2\,E\sum_e \mathbb{E}[\|\mathbf{w}_t^e - \mathbf{w}_t\|^2]\,. \tag{2.22}$$

Using the result derived in Lemma 2 we get,

$$\begin{aligned}
&E\left(\sum_e \mathbb{E}\left[\|\nabla f(\mathbf{w}_t^e) - \nabla f(\mathbf{w}_t)\|^2\right]\right) \\
&\qquad \leq 30\,E^4\,(L + \mu)^2\,\gamma^2\,\left((1 + 2\,B^2\,(P_b + P_n))\|\nabla f(\mathbf{w}_t)\|^2 + b^2 + \sigma^2\right) \\
&\qquad \leq \frac{2\,E^2}{(1 + 2\,B^2\,(P_b + P_n))^2}\,\left((1 + 2\,B^2\,(P_b + P_n))\|\nabla f(\mathbf{w}_t)\|^2 + b^2 + \sigma^2\right)\,.
\end{aligned} \tag{2.23}$$

Now, using Equation (2.23) in Equation (2.21),

$$\mathbb{E}\left[\|\mathbf{g}_t\|^2\right] \leq 2\,E^2\,(2+2P_b\,B^2+P_n\,B^2)\left(1+\frac{2}{(1+2\,B^2\,(P_b+P_n))}\right)\|\nabla f(\mathbf{w}_t)\|^2 + 2\left(1+\frac{(2+2P_b\,B^2+P_n\,B^2)}{(1+2\,B^2\,(P_b+P_n))^2}\right)E^2\,b^2 + 2\left(\frac{1}{2}+\frac{(2+2P_b\,B^2+P_n\,B^2)}{(1+2\,B^2\,(P_b+P_n))^2}\right)E^2\,\sigma^2\,. \tag{2.24}$$

Similarly, let us now focus on bounding the ℓ_2-norm of bias.

$$\begin{aligned}
\|\beta_t\|^2 &= \left\|\sum_{e=0}^{E-1}\nabla\beta_t^e\right\|^2 \\
&\leq E\left(\sum_e\left(P_b\,B^2\|\nabla f(\mathbf{w}_t^e)\|^2+b^2\right)\right) \\
&= P_b\,B^2\,E\left(\sum_e\|\nabla f(\mathbf{w}_t^e)\|^2\right)+E^2\,b^2 \\
&= P_b\,B^2\,E\left(\sum_e\|\nabla f(\mathbf{w}_t^e)-\nabla f(\mathbf{w}_t)+\nabla f(\mathbf{w}_t)\|^2\right)+E^2\,b^2 \\
&\leq 2P_b\,B^2\,E\left(\sum_e\|\nabla f(\mathbf{w}_t^e)-\nabla f(\mathbf{w}_t)\|^2\right)+2P_b\,B^2\,E^2\,\|\nabla f(\mathbf{w}_t)\|^2+E^2\,b^2\,.
\end{aligned}$$

Taking expectation on both sides and using the result derived in (2.23) I get,

$$\|\beta_t\|^2 \leq 2\,P_b\,B^2\,E^2\left(1+\frac{2}{(1+2\,B^2\,(P_b+P_n))}\right)\|\nabla f(\mathbf{w}_t)\|^2 + \frac{2\,E^2}{(1+2\,B^2\,(P_b+P_n))^2}\left(\left(\frac{1}{2}+2\,P_b\,B^2\right)b^2+\sigma^2\right). \tag{2.25}$$

Similarly the upper bound on the second moment of noise ζ_t, I have

$$\mathbb{E}\left[\|\zeta_t\|^2\right] \leq 2\,P_n\,B^2\,E^2\left(1+\frac{2}{(1+2\,B^2\,(P_b+P_n))}\right)\|\nabla f(\mathbf{w}_t)\|^2 + \frac{2\,E^2}{(1+2\,B^2\,(P_b+P_n))^2}\left(b^2+\left(\frac{1}{2}+2P_n\,B^2\right)\sigma^2\right). \tag{2.26}$$

2.9.5 FPS Main Result (Proof of Theorem 2)

In this section, we begin by defining some quantities and notations that will help us deriving the main theorem. We define the quantity: $\tilde{\mathbf{w}}_{t+1}=\mathcal{U}_k(S(\mathbf{w}_{t+1}))$. Here,

$\mathcal{U}_k(S(\cdot))$ represents the unsketching operation. The subscript k denotes the number of top-k coordinates extracted.

As defined in Assumption 1 of the paper, the application specific loss function is $L-$smooth. we denote this application specific loss function as $\ell(\cdot)$. For instance, for a binary classification task, the loss function can be log-loss. Now, our restructured loss function which is formulated by appending a proximal or regularizer term with the leading constant denoted as: μ. This is given by:

$$f(\mathbf{w}, \mathbf{w}^{gb}) = \ell(\mathbf{w}) + \frac{\mu}{2} \left|\left|\mathbf{w} - \mathbf{w}^{gb}\right|\right|^2 , \tag{2.27}$$

where, the iterate $\mathbf{w}^{gb}$ as the last aggregated model parameter vector that was broadcasted by the central server. To simplify, we reduce the notation of $f(\mathbf{w}, \mathbf{w}^{gb})$ to $f(\mathbf{w})$. Here, $\mathbf{w}$ is the current iterate at which the function is being evaluated. Appending such a proximal term preserves the smoothness of the function. Therefore, this new restructured loss function $f(\cdot)$ is $(L + \mu)-$smooth.

We assume that $\gamma \leq \frac{1}{2(L+\mu)}$. Given that $f(\cdot)$ is $(L+\mu)-$smooth,I have that:

$$\begin{aligned}
\mathbb{E}_t[f(\tilde{\mathbf{w}}_{t+1})] &\leq f(\tilde{\mathbf{w}}_t) + \langle \nabla f(\tilde{\mathbf{w}}_t), \mathbb{E}_t[\tilde{\mathbf{w}}_{t+1} - \tilde{\mathbf{w}}_t] \rangle + \frac{(L+\mu)}{2} \mathbb{E}_t \left[\|\tilde{\mathbf{w}}_{t+1} - \tilde{\mathbf{w}}_t\|^2 \right] \\
&= f(\tilde{\mathbf{w}}_t) - \langle \nabla f(\tilde{\mathbf{w}}_t), \gamma \, \mathbb{E}_t[\mathbf{g}_t] \rangle + \frac{(L+\mu)}{2} \mathbb{E}_t \left[\|\gamma \, \mathbf{g}_t\|^2 \right] \\
&= f(\tilde{\mathbf{w}}_t) - \gamma \langle \nabla f(\mathbf{w}_t), \mathbb{E}_t[\mathbf{g}_t] \rangle + \langle \nabla f(\mathbf{w}_t) - \nabla f(\tilde{\mathbf{w}}_t), \gamma \, \mathbb{E}_t[\mathbf{g}_t] \rangle \\
&\qquad + \frac{(L+\mu)}{2} \gamma^2 \mathbb{E}_t \left[\|\mathbf{g}_t\|^2 \right] \\
&\overset{(a)}{\leq} f(\tilde{\mathbf{w}}_t) - \gamma \langle \nabla f(\mathbf{w}_t), \nabla f(\mathbf{w}_t) + \beta_t \rangle + \langle \nabla f(\mathbf{w}_t) - \nabla f(\tilde{\mathbf{w}}_t), \gamma \, \mathbb{E}_t[\mathbf{g}_t] \rangle \\
&\qquad + \gamma^2 (L+\mu) \left(\|\nabla f(\mathbf{w}_t) + \beta_t\|^2 + \mathbb{E}_t \left[\|\zeta_t\|^2 \right] \right) \\
&\overset{(b)}{\leq} f(\tilde{\mathbf{w}}_t) + \frac{\gamma}{2} \left(-2 \langle \nabla f(\mathbf{w}_t), \nabla f(\mathbf{w}_t) + \beta_t \rangle + \|\nabla f(\mathbf{w}_t) + \beta_t\|^2 \right) \\
&\qquad + \langle \nabla f(\mathbf{w}_t) - \nabla f(\tilde{\mathbf{w}}_t), \gamma \, \mathbb{E}_t[\mathbf{g}_t] \rangle + \gamma^2 (L+\mu) \left(\mathbb{E}_t \left[\|\zeta_t\|^2 \right] \right) \\
&= f(\tilde{\mathbf{w}}_t) + \frac{\gamma}{2} \left(-\|\nabla f(\mathbf{w}_t)\|^2 + \|\beta_t\|^2 \right) + \langle \nabla f(\mathbf{w}_t) - \nabla f(\tilde{\mathbf{w}}_t), \gamma \, \mathbb{E}_t[\mathbf{g}_t] \rangle \\
&\qquad + \gamma^2 (L+\mu) \left(\mathbb{E}_t \left[\|\zeta_t\|^2 \right] \right), \qquad (2.28)
\end{aligned}$$

where, inequality (a) is a consequence of using Young's inequality. Inequality (b) is a direct consequence of using the assumption $\gamma \leq \frac{1}{2\,(L+\mu)}$ from Lemma 2. To keep our analysis visually easy to follow I abbreviate the quantity $\frac{1}{1+2\,B^2\,(P_b+P_n)}$ as H.
Continuing on with our proof from Equation (2.28) and utilizing the second moment

bounds from Equations (2.24) , (2.25) and (2.26) we get:

$$
\begin{aligned}
&\mathbb{E}_t[f(\tilde{\mathbf{w}}_{t+1})] \\
&\leq f(\tilde{\mathbf{w}}_t) - \left(\frac{\gamma}{2} - \frac{\gamma\, P_b\,(1+2\,H)\,E^2\,B^2}{2} - 2\,P_n\,(L+\mu)\,(1+2\,H)\gamma^2\,E^2\,B^2\right)\,\|\nabla f(\mathbf{w}_t)\|^2 \\
&\quad + 2\,E^2\,H^2\left(\left(\frac{1}{2}+2\,P_b\,B^2\right)\,\gamma + \gamma^2\,(L+\mu)\right)\,b^2 \\
&\quad + 2\,E^2\,H^2\left(\gamma + \left(\frac{1}{2}+2\,P_n\,B^2\right)\,\gamma^2\,(L+\mu)\right)\,\sigma^2 \\
&\quad + \mathbb{E}_t[\langle (L+\mu)\,(\mathbf{w}_t - \tilde{\mathbf{w}}_t), \gamma\,\mathbf{g}_t\rangle] \\
&\overset{(d)}{\leq} f(\tilde{\mathbf{w}}_t) - \left(\frac{\gamma}{2} - \frac{\gamma\;P_b\,(1+2\,H)\,E^2\,B^2}{2} - 2\,P_n\,(L+\mu)\,(1+2\,H)\gamma^2\,E^2\,B^2\right)\,\|\nabla f(\mathbf{w}_t)\|^2 \\
&\quad + 2\,E^2\,H^2\left(\left(\frac{1}{2}+2\,P_b\,B^2\right)\,\gamma + \gamma^2\,(L+\mu)\right)\,b^2 \\
&\quad + 2\,E^2\,H^2\left(\gamma + \left(\frac{1}{2}+2P_n\,B^2\right)\,\gamma^2\,(L+\mu)\right)\,\sigma^2 \\
&\quad + \frac{(L+\mu)^2}{2}\,\mathbb{E}_t\left[\|\mathbf{w}_t - \tilde{\mathbf{w}}_t\,\|^2\right] + \frac{\gamma^2}{2}\,\mathbb{E}_t\left[\|\mathbf{g}_t\|^2\right] \\
&\leq f(\tilde{\mathbf{w}}_t) + \frac{(L+\mu)^2}{2}\mathbb{E}_t\left[\|\mathbf{w}_t - \tilde{\mathbf{w}}_t\|^2\right] - \rho(\gamma)\|\nabla f(\mathbf{w}_t)\|^2 \\
&+ 2\,E^2H^2\left(\left(\frac{1}{2}+2\,P_b\,B^2\right)\gamma + \gamma^2\,(L+\mu) + \left(\frac{1}{H^2} + (2+2P_bB^2+P_nB^2)\right)\gamma^2\right)b^2 \\
&+ 2E^2H^2\left(\gamma + \left(\frac{1}{2}+2P_nB^2\right)\gamma^2(L+\mu) + \left((2+2P_bB^2+P_nB^2) + \frac{1}{2H^2}\right)\gamma^2\right)\sigma^2. \qquad (2.29)
\end{aligned}
$$

Here, we define the quantity $\rho(\gamma)$ as:

$$
\begin{aligned}
\rho(\gamma) &= \frac{1 - P_b\,(1+2\,H)\,E^2\,B^2}{2} - 2\,P_n\,(L+\mu)\,(1+2\,H)\gamma\,E^2\,B^2 \\
&\qquad - \gamma\,E^2\,(2+2P_b\,B^2+P_n\,B^2)\,(1+2\,H) \\
&= \frac{1 - P_b\,(1+2\,H)\,E^2\,B^2}{2} \\
&\qquad - \gamma\,(2+2P_b\,B^2 + (2(L+\mu)+1)\,P_n\,B^2)\,(1+2\,H)\;E^2\,. \qquad (2.30)
\end{aligned}
$$

We want the above defined quantity ($\rho(\gamma)$) to be greater than 0. This provides us with a bound on the learning rate and it is given by:

$$\gamma < \frac{1 - P_b\,(1 + 2\,H)\,E^2\,B^2}{2(2 + 2P_b\,B^2 + (2(L+\mu)+1)\,P_n\,B^2)\,(1+2\,H)\;E^2}$$

Since $H \leq 1$ we get:

$$\gamma \leq \frac{1 - 6\,P_b\,E^2\,B^2}{12(1 + P_b\,B^2 + (L+\mu+1)\,P_n\,B^2)\,E^2}\,. \tag{2.31}$$

Now averaging from 0 to T on both sides and plugging the bound for residual term (highlighted in red in Equation (2.29)) by Lemma 1 the following holds with probability $1-\delta$:

$$\begin{aligned}
\frac{1}{T+1}\sum_{t=0}^{T} \gamma\,\rho(\gamma)\,\|\nabla f(\mathbf{w}_t)\|^2 \leq{}& \frac{|f(\mathbf{w}_0) - f(\mathbf{w}^*)|}{(T+1)} + \left(\frac{1}{c} + \frac{(k+1)^{1-2p} - d^{1-2p}}{2p-1}\right) W \\
&+ 2\,E^2\,H^2\left(\left(\frac{1}{2} + 2P_b\,B^2\right)\gamma + \gamma^2\,(L+\mu) + \left(\frac{1}{H^2} + (2 + 2P_b\,B^2 + P_n\,B^2)\right)\gamma^2\right) b^2 \\
&+ 2\,E^2\,H^2\left(\gamma + \left(\frac{1}{2} + 2P_n\,B^2\right)\gamma^2\,(L+\mu) + \left((2 + 2P_b\,B^2 + P_n\,B^2) + \frac{1}{2\,H^2}\right)\gamma^2\right)\sigma^2\,.
\end{aligned}$$

Then using the fact that $H \leq 1$ and rearranging terms,

$$\begin{aligned}
&\frac{1}{T+1}\sum_{t=0}^{T} \rho(\gamma)\|\nabla f(\mathbf{w}_t)\|^2 \\
&\leq \frac{|f(\mathbf{w}_0) - f(\mathbf{w}^*)|}{\gamma(T+1)} + \left(\frac{1}{c} + \frac{(k+1)^{1-2p} - d^{1-2p}}{2p-1}\right) W \\
&\quad + 2E^2\left(\left(\frac{1}{2} + 2P_bB^2\right) + \gamma(L+\mu) + \big(1 + (2 + 2P_bB^2 + P_nB^2)\big)\,\gamma\right) b^2 \\
&\quad + 2E^2\left(1 + \left(\frac{1}{2} + 2P_nB^2\right)\gamma(L+\mu) + \left((2 + 2P_bB^2 + P_nB^2) + \frac{1}{2}\right)\gamma\right)\sigma^2 \\
&\leq \frac{|f(\mathbf{w}_0) - f(\mathbf{w}^*)|}{\gamma(T+1)} + \left(\frac{1}{c} + \frac{(k+1)^{1-2p} - d^{1-2p}}{2p-1}\right) W \\
&\quad + 2E^2\left(1 + \gamma(L+\mu+1)(3 + 2P_bB^2 + 2\,P_nB^2)\right)\sigma^2 \\
&\quad + 2E^2\left(1 + 2P_bB^2 + \gamma(3 + L + \mu + 2P_bB^2 + 2P_nB^2)\right) b^2.
\end{aligned} \tag{2.32}$$

Chapter 3

ADAPTIVE CLIENT SELECTION IN LARGE-SCALE FEDERATED LEARNING

3.1 Introduction

As stated in the previous chapter, FL has emerged as a powerful paradigm for collaborative machine learning, enabling multiple parties to train models without sharing raw data. However, the heterogeneous distribution of data across clients and the need for frequent communication pose significant challenges to efficiency and scalability. In the previous chapter, a novel approach to reducing communication overhead by transmitting compressed estimates and incorporating local training iterations, all in the presence of heterogeneous data distributions across clients was discussed.

Another promising direction in the efforts to reduce communication and computation overhead is the development of adaptive client selection strategies. Traditional federated learning approaches often involve uniform sampling of clients, which can be inefficient when data is heterogeneously distributed. Our work builds on the observation that in such scenarios, only a subset of clients significantly contributes to the global model's learning process. By prioritizing these informative clients, we can potentially reduce both communication and computational costs while maintaining model performance.

Drawing inspiration from active learning, we propose an Uncertainty-based Active Client Selection for Federated Learning (UACS-FL) approach. This method employs entropy as an information-theoretic measure of uncertainty to identify clients with the most informative data distributions. At each federated learning epoch, our strategy

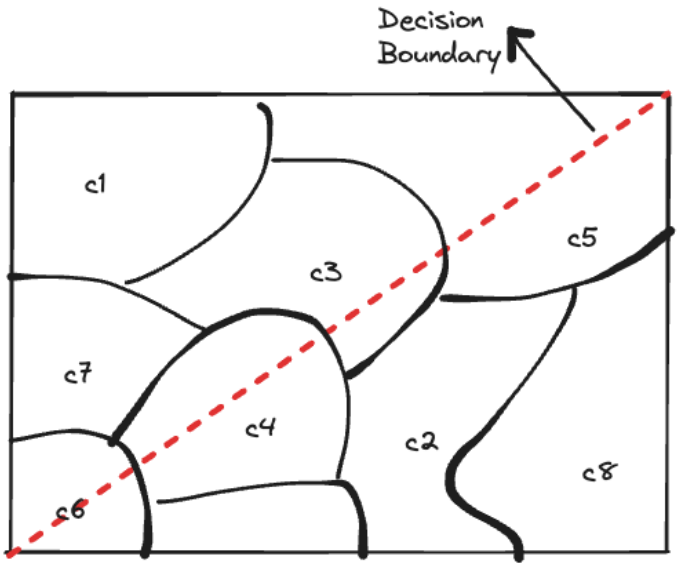


Figure 13. Illustration of a simple classification task where only a handful of clients possess valuable information to learn the decision boundary efficiently.

selects clients with the highest measured uncertainty, focusing computational resources on areas of the data space that are likely to yield the most significant improvements to the global model. To motivate the efficacy of the proposed approach (UACS-FL) consider the following illustration of how data is distributed across different clients. From Figure below it is clear that only a fraction of clients (c_3, c_4, c_5 and c_6) contribute significantly to the learning of decision boundary in red.

Recent studies have explored both biased and unbiased client participation strategies in federated learning. While unbiased approaches aim to ensure fair representation across all clients, biased participation has shown promise in scenarios with highly skewed data distributions. Our UACS-FL method intentionally introduces a bias towards more uncertain clients, potentially accelerating convergence and improving model performance in heterogeneous settings.

By combining elements of active learning with federated optimization, our approach

aims to strike a balance between model performance, communication efficiency, and computational cost. This work contributes to the ongoing efforts in the federated learning community to develop more adaptive and efficient training protocols for large-scale distributed systems.

3.2 Related Works

Most of the recent works mentioned above assume full client participation, meaning all nodes are involved in every training round. However, in practice, only a small fraction of clients participate in each round, which can intensify the negative effects of data heterogeneity. While some convergence guarantees for full participation and techniques to address heterogeneity can be adapted to partial participation X. Li et al. 2020, these adaptations are limited to unbiased client participation, where each client's contribution is proportional to its dataset size. In Ruan et al. 2020, the authors study convergence with flexible device participation, allowing devices to freely join or leave the process or send incomplete updates to the server. However, adaptive client selection, which takes into account the progress of each client during training, remains poorly understood.

Understanding biased client selection strategies is important because they can significantly accelerate error convergence and improve communication efficiency in heterogeneous environments by preferentially selecting clients with higher local loss values, as shown in this paper. This concept has been explored in a few recent empirical studies Goetz et al. 2019; Laguel et al. 2020; Ribero et al. 2020. Nishio et al. 2019 proposed grouping clients based on hardware and wireless resources to save communication resources, while Goetz et al. 2019 suggested client selection based on local loss, and Ribero et al. 2020 proposed using the progression of clients' weights.

However, these methods are limited to empirical demonstrations without a detailed analysis of how selection bias affects convergence speed.

Another relevant body of work Jiang et al. 2019; Katharopoulos et al. 2018; Shah et al. 2020; Salehi et al. 2018 utilizes biased selection or importance sampling of data to accelerate the convergence of classic centralized SGD. These methods advocate for selecting samples with the highest loss or the largest gradient norm (which we use as a benchmark in our experiments) for the next iteration of SGD.

3.3 Problem Setup and Notation

The notation used throughout the development of the UACS-FL algorithm is described in Table 4:

Notation	**Description**
$\mathbf{w}_t$	Global model parameters at communication round t.
α_i	Selection metric for client i, representing the entropy-based uncertainty score used for client selection.
M	Total number of clients in the federated network.
m	Number of clients selected to participate in each communication round.
$\mathcal{D}_i$	Local dataset held by client i.

Table 4. Notation used in UACS-FL framework.

UACS-FL builds on the FedAvg framework McMahan et al. 2016, incorporating an active learning (AL) metric based on entropy to prioritize clients for participation in each training round. The method aims to enhance communication efficiency while maintaining model performance across diverse client data distributions.

The global objective function is defined as:

$$\min_{\mathbf{w}} \left[f(\mathbf{w}) := \sum_{i=1}^{M} c_i f_i(\mathbf{w}) \right],$$

where, each client i has a local objective function $\ell_i(\mathbf{w})$, and the scalar c_i represents the weight assigned to client i in the aggregation. To simplify, we assume equal weighting across clients, setting $c_i = \frac{1}{M}$ for all $i \in [M]$.

To account for partial client participation at each communication round t, we modify the global objective to include only the subset of clients that participate in that round. Let $S_t \subseteq [M]$ denote the subset of selected clients at round t. The revised objective function can thus be formulated as:

$$\min_{\mathbf{w}} \left[f(\mathbf{w}) := \frac{1}{|S_t|} \sum_{i \in S_t} f_i(\mathbf{w}) \right],$$

where $S_t \subseteq [M]$: the subset of clients selected to participate in round t. $|S_t|$: the cardinality of S_t, representing the number of clients selected in this round.

The normalization factor $\frac{1}{|S_t|}$ is included to ensure that the contribution of the objective remains consistent across rounds, even when the number of participating clients varies. This formulation retains the structure of the original optimization objective while allowing for dynamic selection of clients at each round based on selection criteria, such as data heterogeneity or communication efficiency. This optimization is performed iteratively until it reaches a stationary point.

3.4 Proposed Approach: Uncertainty-Aware Client Selection Federated Learning

The following outlines the detailed steps of the UACS-FL method. The algorithm is provided in Algorithm 3.

Initialization. The server initializes the global model parameters, denoted $\mathbf{w}_0$, which serve as the starting point for the federated learning process. Additionally, the server specifies the number of clients m to be selected at each round. This choice

of m reflects a balance between reducing communication overhead and ensuring a representative set of clients participate in each round.

After initialization the iterative learning starts where each round t comprises three main steps: *Active Client Selection*, *Client-Side Computation*, and *Server-Side Aggregation*.

Active Client Selection. In each round, the server selects a subset of clients based on their computed selection metric, denoted α_i. This process aims to capture the most informative clients using an epsilon-greedy approach, which alternates between exploration (random client selection) and exploitation (selection based on highest metric values). The steps are as follows:

1. **Metric Computation**: Each client $i \in \{1, 2, \ldots, M\}$ receives the global model parameters $\mathbf{w}_t$ from the server. Using its local dataset $\mathcal{D}_i$, each client computes an entropy-based metric α_i, which quantifies the uncertainty in its local data relative to the global model. This metric acts as a proxy for client informativeness, where higher values of α_i indicate a greater potential contribution to the global model. The entropy-based metric for a sample $(x, y) \in \mathcal{D}$ is defined as:

$$H(x) = -\sum_{\ell} P_\theta(y_\ell|x) \log P_\theta(y_\ell|x), \tag{3.1}$$

where y_ℓ ranges over all possible labels and $P_\theta(y_\ell|x)$ represents the probability of label y_ℓ given sample x under model parameters θ.

We calculate the entropy metric for client c_i as follows:

$$\alpha_i = \frac{-\sum_{n=1}^{|\mathcal{D}_i|} \sum_{ell} P_\theta(y_\ell^{i,n}|x^{i,n}) \log P_\theta(y_\ell^{i,n}|x^{i,n})}{|\mathcal{D}_i|}, \tag{3.2}$$

where $x^{i,n}$ and $y_\ell^{i,n}$ denote the n-th sample and its corresponding label in $\mathcal{D}_i$, respectively.

2. **Metric Submission**: Each client sends its computed selection metric α_i to the server, allowing the server to make an informed choice regarding which clients to engage in this round.
3. **Client Selection Using Epsilon-Greedy Strategy**:
    - **Exploration**: With probability ϵ, the server randomly selects m clients from the entire set $\{1, 2, \ldots, N\}$, disregarding the computed α_i values. This strategy prevents overfitting to particular clients by including a broader range of data sources over time.
    - **Exploitation**: With probability $1 - \epsilon$, the server selects the top m clients with the highest α_i values. This step prioritizes clients whose data is expected to yield the most informative updates, based on their high entropy metrics.

Client-Side Computation. Each selected client $i \in S_t$ proceeds to perform local updates on the global model using its local dataset $\mathcal{D}_i$. This step includes:

1. **Local Model Update**: Each client initializes its model with the received global parameters $\mathbf{w}_t$. Subsequently, the client performs a series of gradient descent steps to minimize the local objective function based on D_i, resulting in updated local model parameters $\mathbf{w}_{t+1}^i$. This local training step adapts the global model to the client's data distribution.

Server-Side Aggregation. After receiving the updated parameters from the selected clients, the server performs the following steps:

1. **Aggregation of Local Models**: The server aggregates the local model updates received from the selected clients S_t. The aggregation is performed by taking

an average of the local models, weighted equally across the selected clients, to produce the updated global model parameters:

$$\mathbf{w}_{t+1} = \frac{1}{m} \sum_{i \in S_t} \mathbf{w}_i^{(t+1)}$$

This aggregation step enables the server to combine the insights obtained from each selected client, thus refining the global model to better represent the participating clients' data.

2. **Broadcast of Updated Global Model**: The server broadcasts the aggregated model $\mathbf{w}_{t+1}$ to all clients, completing one round of federated learning. This updated global model serves as the starting point for the next round.

Algorithm 3 UACS-FL: Uncertainty-Aware Client Selection for Federated Learning

Initialization:
  Server initializes the global model parameters $\mathbf{w}_0$.
  Server sets the number of selected clients m per round.
for each round $t = 1, 2, \ldots$ **do**
  Active Client Selection:
  for each client $i \in \{1, 2, \ldots, M\}$ **do**
    Client receives global model parameters $\mathbf{w}_t$ from the server.
    Client computes the selection metric α_i based on its local model (e.g., entropy).
    Client sends α_i to the server.
  end for
  Server selects a set S_t of m clients using an epsilon-greedy approach:
    Generate a random number $r \in [0, 1]$.
  if $r \leq \epsilon$ **then** ▷ Exploration step
    Randomly select m clients from $\{1, 2, \ldots, n\}$ without considering α_i.
  else ▷ Exploitation step
    Select the top m clients with the highest α_i values.
  end if
  Client-Side Computation:
  for each client $i \in S_t$ **do**
    Client updates its local model using its local dataset $\mathcal{D}_i$ to compute the local model parameters $\mathbf{w}_{t+1}^i$.
  end for
  Server-Side Aggregation:
  Server aggregates the parameters from the selected m clients:

$$\mathbf{w}_{t+1} = \frac{1}{m} \sum_{i \in S_t} \mathbf{w}_{t+1}^i$$

  Server broadcasts the updated global model parameters $\mathbf{w}_{t+1}$ to all clients.
end for

3.5 Experimental Studies

We consider two variants of UACS-FL for client selection:

1. **Entropy + Epsilon-Greedy:** Epsilon-greedy parameter $\epsilon = 0.1$. In our plots, we label this as Entropy+Eps-Greedy.

2. **Entropy Based (Pure Exploitation):** Epsilon-greedy parameter $\epsilon = 0 \implies$ pure exploitation. In our plots, we label this as Entropy Based.

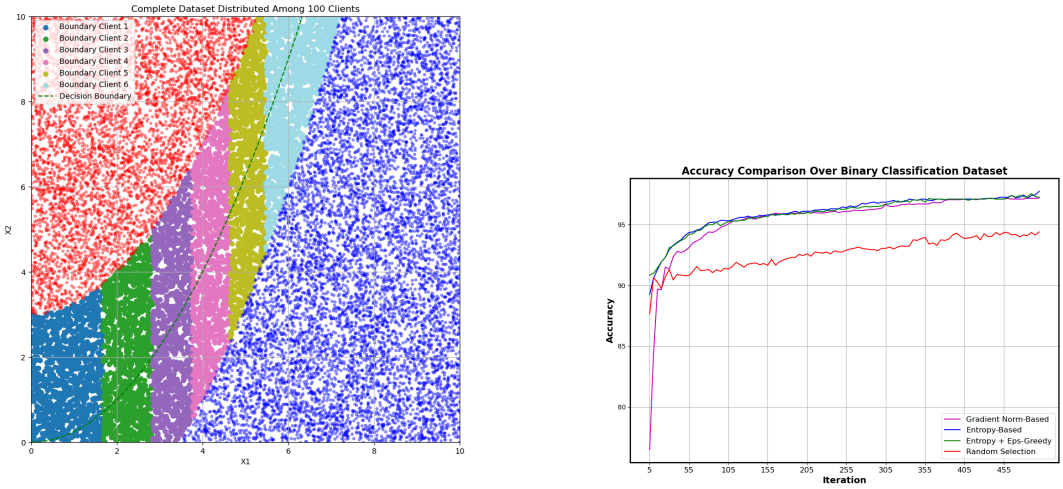
We compare our proposed approach against the following baseline client selection strategies:

- **Random Client Selection:** Clients are selected at random each round. In our plots, we label this as Random Selection.
- **Gradient Norm-Based Selection:** Clients are chosen based on the largest gradient norm values. In our plots, we label this as Gradient Norm-Based.

We evaluate the performance of our algorithm and the baseline approaches on three distinct data distribution settings within a binary classification task. In each of these experiments, the number of clients contributing effectively to the learning of the decision boundary is limited (around 6) compared to the total number of clients (100). Across all data distribution plots, the dotted green line indicates the decision boundary.

3.5.1 Experimental Study

In this experiment, only 6 out of the 100 clients have samples positioned around the decision boundary. The sample distribution for each of these 6 clients is illustrated below. The remaining 94 clients each have access to samples corresponding to only one class (either red or blue). This is shown in Figure 14a. The model is a simple 2 layer neural network. In Figure 14b it can be seen that random client selection consistently underperforms in comparison to the proposed method. Furthermore, while the gradient norm-based client selection achieves comparable results, the proposed method demonstrates superior performance, notably achieving faster convergence.



(a) (b)

Figure 14. For experiment 1: (a) data distribution across different clients. Only a fraction of clients are close to the decision boundary, (b) Plotting accuracy of UACS-FL with other competing state of the art algorithms like gradient-norm based and random client selection.

3.6 Conclusion

In this chapter, we introduced the UACS-FL framework to address the challenge of selecting clients in federated learning environments where full participation is either infeasible or inefficient. By focusing on client selection, UACS-FL leverages uncertainty-based and adaptive strategies to ensure that only the most informative clients contribute to each aggregation cycle, thereby enhancing both model accuracy and convergence rates. Through preliminary experimentation, we demonstrated that UACS-FL not only reduces communication overhead but also improves model performance by prioritizing clients with diverse and representative data distributions. Future work may explore how UACS-FL can be adapted to further account for additional constraints, such as energy usage, improving communication efficiency by

utilizing compression technique, to broaden its applicability in emerging federated learning scenarios.

Chapter 4

STOCHASTIC LINEAR BANDITS WITH LIMITED OBSERVABILITY

4.1 Introduction

The field of sequential decision-making under uncertainty has seen significant advancements in recent years, particularly in the domain of stochastic linear bandits Yasin Abbasi-Yadkori et al. 2011; Dani et al. 2008; Rusmevichientong et al. 2010. In this paradigm, an agent is tasked with sequentially selecting the optimal action from a set of action vectors presented at each time instant, with the goal of maximizing cumulative rewards over time. The stochastic linear bandit model has found widespread applications in various domains, including online advertising L. Li et al. 2010, recommendation systems S. Li et al. 2016, and clinical trials Durand et al. 2018.

In stochastic linear bandits (SLB) Yasin Abbasi-Yadkori et al. 2011, an agent sequentially selects the optimal action from a set of action vectors presented at each time instant. At each round, the agent chooses an action, and the environment returns a stochastic reward with some expected value. The agent's goal is to maximize cumulative rewards over T interactions. In SLB, actions are represented as d-dimensional vectors. Throughout the interaction, the agent builds a model of the environment, making decisions not only to maximize current rewards but also to explore other actions, aiming to improve its estimate of the unknown linear function for higher future rewards—a process known as the exploration-exploitation trade-off. During exploration, the agent may make mistakes by selecting suboptimal actions, accumulating regret, which is the cumulative cost of these mistakes. The agent's objective is to design a strategy to minimize this regret.

A widely used approach is the optimism in the face of uncertainty (OFU) principle, where the agent estimates the environment's model within a confidence interval and constructs a plausible set of models. It then chooses the most optimistic model from this set, guiding the next round's decision-making. For general SLB problems, Yasin Abbasi-Yadkori et al. 2011 introduced the OFUL algorithm, deriving a regret upper bound of $\mathcal{O}(d\sqrt{T})$, which matches the lower bound up to a logarithmic factor. However, in high-dimensional settings, especially when d is large, these regret bounds become practically infeasible.

With the rise of large-scale datasets, action vectors have become increasingly high-dimensional. Real-world problems, however, often exhibit hidden low-dimensional structures, suggesting that action vectors lie in a lower-dimensional space. For instance, in recommendation systems, items are represented by detailed, hand-engineered feature vectors, making d intractably large. Yet, not all features contribute effectively to the recommendation task.

Despite this, traditional SLB works have not fully exploited the inherent low-dimensional structures of action vectors. Recent approaches, such as Lale et al. 2019, leverage dimension reduction and subspace recovery techniques to harness these low-dimensional structures, improving decision-making within the SLB framework.

Traditionally, it is assumed that the agent has access to a fully observed decision set, enabling comprehensive inference and optimal action recommendation. However, the high dimensionality often necessitates strategies to reduce data storage costs. A common approach is sub-sampling the action vectors, introducing erasures. Alternatively, assuming a noisy observation model can naturally sparsify the high-dimensional action vectors.

To address the challenges of high-dimensional and incomplete data in stochastic

linear bandits (SLB), we introduce a novel algorithm tailored for scenarios with missing action vector entries. Our approach optimizes action selection by leveraging subspace recovery techniques, specifically using PCA with missing entries. By identifying a low-dimensional m-dimensional subspace, we effectively reduce the complexity of the SLB problem while maintaining high performance. Our analysis shows that the proposed method achieves a regret bound that depends on the subspace dimension m and the probability of observing entries in action vectors p. This result highlights a trade-off between these parameters: as the subspace dimension m increases, the model's complexity grows, potentially increasing regret. Conversely, when $m \ll d$, subspace recovery facilitates faster learning of the underlying linear function, resulting in smaller regret. However, a lower probability p of observing entries reduces the available information, which in turn can negatively affect performance by accumulating higher regret. Notably, our regret is independent of the ambient dimension d, making it well-suited for high-dimensional problems. By focusing on the intrinsic low-dimensional structure, our algorithm ensures efficient learning even with partial observations, highlighting its robustness and adaptability.

This result is particularly significant in real-world applications where data is often incomplete or partially observed, and direct high-dimensional processing is computationally prohibitive. Our method demonstrates that it is possible to achieve sub-linear regret across various values of p, confirming its effectiveness in scenarios with limited observability. By reducing dimensionality through subspace recovery, we not only enhance the efficiency of decision-making in SLB but also provide a scalable solution that remains robust as the problem's dimensionality increases.

4.2 Related Work

The study of linear bandit problems extends to various algorithms and environment settings Dani et al. 2008; Rusmevichientong et al. 2010; L. Li et al. 2010. Kleinberg et al. 2010 studies the class of problems when the decision set changes time to time, while Dani et al. 2008 studies this problem when the decision set provides a set of fixed actions. Further analysis in the area extend these approaches to classes where there are more structures in the problem setup. In traditional decision-making problems, where hand engineered feature representations are provided, sparsity in the linear function is a valid structure. Sparsity, as the key in high-dimensional conventional structured linear bandits, conveys series of successes in classical settings Y. Abbasi-Yadkori et al. 2012; Carpentier et al. 2012. In fact Y. Abbasi-Yadkori et al. 2012 show how to exploit this additional structure and design a practical algorithm with regret of $\mathcal{O}(\sqrt{sdT})$, where s is the sparsity level of the true underlying linear function. Under slightly stronger assumptions, Carpentier et al. 2012 show the theory of compressed sensing can provide a tighter bound of $\mathcal{O}(s\sqrt{T})$.In recommendations systems, where a set of users and items are given, Gopalan et al. 2016 consider the low-rank structure of the user-item preference matrix and provide an algorithm which exploits this further structure.

On the other hand, subspace recovery and dimension reduction problems are well studied in the literature. Subspace estimation is a fundamental problem in signal processing and machine learning with applications in areas such as dimensionality reduction, anomaly detection, and recommendation systems. Classical methods like principal component analysis (PCA) estimate subspaces from fully observed dataJolliffe et al. 2016. However, many real-world scenarios involve incomplete or partially observed data.

Several algorithms have been proposed for subspace estimation with missing data. Oja's method Oja 1982 is a classical online algorithm that can be adapted to handle missing entries. More recent approaches include GROUSE (Grassmannian Rank-One Update Subspace Estimation) Balzano et al. 2010 and PETRELS (Parallel Subspace Estimation and Tracking by Recursive Least Squares) Chi et al. 2013, which are specifically designed for the streaming setting with partial observations. More recently, there has been interest in high-dimensional analysis of subspace estimation algorithms. Liang et al. Liang et al. 2019 provided a unified framework for analyzing several algorithms including Oja's method, GROUSE and PETRELS in the high-dimensional regime. Their work characterized the asymptotic dynamics of these algorithms through deterministic ordinary differential equations. Other related lines of work include robust PCA Candès et al. 2011, which aims to recover low-rank subspaces in the presence of sparse corruptions, and subspace tracking T. Yang et al. 2015, which deals with time-varying subspaces. Matrix completion Candès et al. 2009 is also closely related, as subspace estimation can be viewed as a special case of low-rank matrix recovery.

To the best of our knowledge, the only work in the literature that considers hidden low-dimensional subspace assumptions on actions and/or unknown weight vectors in stochastic linear bandits (SLB) is Lale et al. 2019. However, there exists no work which looks at partial observability of actions vectors in low-dimensional setting which is exactly the focus of this work.

4.3 Problem Setup

For a natural number $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, 2, \ldots, n\}$.Let $T \in \mathbb{N}$ be the time horizon of the decision-making game. Let $\mathbf{U}$ be a $d \times m$ fixed but unknown orthonormal matrix with $m < d$; span($\mathbf{U}$) defines an $m-$dimensional subspace in

$\mathbb{R}^d$. As discussed in Section 2.1, we will suppose that the (partially observed) action vectors available to the agent lie in span($\mathbf{U}$). That is, at every round $t \in [T]$, we will suppose that a fixed number $K \in \mathbb{N}$ of "ideal" actions are drawn according to

$$X_{t,i} = \mathbf{U}\mathbf{\Lambda}Z_{t,i}, \quad i \in [K]. \tag{4.1}$$

Each coordinate k of the random vector $Z_{t,i} \in \mathbb{R}^m$ is drawn from a fixed distribution p_Z such that $\mathbb{E}\left[[Z_{t,i}]_k\right] = 0$, $\text{Var}\left[[Z_{t,i}]_k\right] = 1$, and $|[Z_{t,i}]_k| \leq M$, with probability 1. $\mathbf{\Lambda} \in \mathbb{R}^{m \times m}$ a fixed diagonal "loading" matrix.

Under the generative model in Equation (4.1), we can upper bound the $2-$norm of any action vector as $\|X_{t,i}\|_2 \leq \lambda_1(\sqrt{m})M \triangleq L$, where $\lambda_1 = \max_i |\mathbf{\Lambda}_{ii}|$.

Notation	**Description**
d	Dimension of the observation space
m	Subspace dimension
p	Probability of observing a coordinate
μ	Incoherence parameter of orthonormal basis
T	Total number of rounds
K	Number of arms received per round
λ_1	Largest eigenvalue in $\mathbf{\Lambda}-$ a fixed diagonal "loading" matrix.
λ_m	m^{th} largest eigenvalue in $\mathbf{\Lambda}$
$X_{t,i}$	Full observed i^{th} action vector at time t
$\dot{X}_{t,i}$	Partially observed i^{th} action vector at time t
θ^*	True model parameter
$\|.\|_2$	Euclidean norm for vectors and operator norm for matrices
$\mathcal{O}(\cdot)$	Big-O notation for asymptotic scaling
$\tilde{\mathcal{O}}(\cdot)$	Big-O notation ignoring logarithmic factors

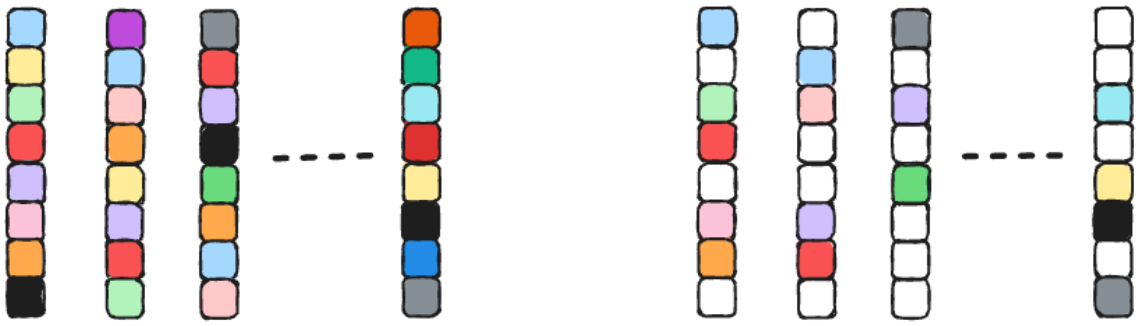
Table 5. Notations and their respective descriptions.

At each time step $t \in [T]$, for each $i \in [K]$, we partially observe the ideal action vector $X_{t,i}$. That is, for each $j \in [d]$ we sample $s_{t,i}^{(j)} \overset{\text{iid}}{\sim} \text{Bernoulli}(p)$. The partially

observed action vector is then given by

$$\dot{X}_{t,i}^{(j)} = X_{t,i}^{(j)} \cdot s_{t,i}^{(j)}. \tag{4.2}$$

This is illustrated in Figure 15. At each round t, we are presented with the following



(a) (b)

Figure 15. The sequence of action vectors $\{X_t\}_{t=1}^T$ drawn from a generative model specified in Equation (4.1) with unknown m-dimensional subspace $\mathbf{U}$ in panel (a) is only partially observed on random index sets $\{\Omega_t\}_{t=1}^T$. The index sets at each time are generated according to Equation (4.2). That is, only incomplete action vectors $\{\dot{X}_t\}_{t=1}^T$ are accessible in panel (b), where the white entries are missing.

decision set $D_t = \{\dot{X}_{t,1}, \ldots, \dot{X}_{t,K}\}$ and the algorithm chooses an action vector and observes a reward r_t such that

$$r_t = X_t^T \theta^* + \eta_t, \qquad \forall t \in [T] \tag{4.3}$$

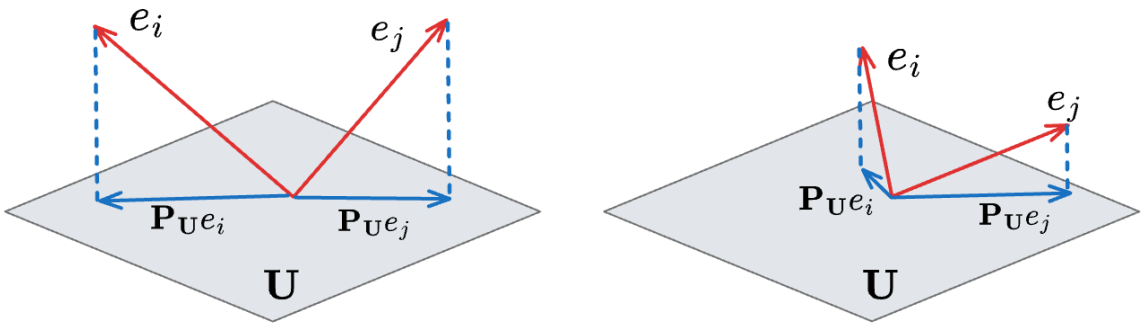
where, $\theta^* \in \text{span}(\mathbf{U})$ is an unknown parameter vector, and η_t is sub-Gaussian noise (defined in Definition 4).

Definition 3 (Incoherence) *The incoherence parameter μ of $\mathbf{U}$, is defined as:*

$$\mu(\mathbf{U}) = \mu := \sqrt{\frac{d}{m}} \max_i \|\mathbf{U}_i\|_2,$$

where $\mathbf{U}_i$ is the i-th row of $\mathbf{U}$.

Matrices with small incoherence parameter μ are referred to as *incoherent matrices*. These matrices are "spread out," meaning that no single row of $\mathbf{U}$ dominates the others in terms of its norm. The incoherence parameter plays a critical role in understanding the sample complexity of various problems with missing entries Candès et al. 2009. If a matrix is not incoherent (i.e., it has a large μ), a few entries contain most of the matrix's norm (or energy). Consequently, recovering such matrices or their principal components becomes significantly more challenging. This is illustrated in Figure 16.



(a) Small incoherence (b) Large incoherence

Figure 16. Illustration of the incoherence parameter $\mu(\mathbf{U})$. The value of $\mu(\mathbf{U})$ is small when all the standard basis vectors e_i have approximately the same projections onto the subspace $\mathbf{U}$, as shown in (a); and $\mu(\mathbf{U})$ is large if $\mathbf{U}$ is too aligned with certain standard basis vector, as shown in (b).

Definition 4 (Sub-gaussian Noise) *For all $t \in [T]$, η_t is conditionally $R-$sub-Gaussian, there exists a constant $R \geq 0$ such that, $\forall\ \lambda \in \mathbb{R}$, $\mathbb{E}[e^{\lambda \eta_t} | X_{1:t}, \eta_{1:t-1}] \leq e^{\frac{\lambda^2 R^2}{2}}$.*

The goal of the agent is to maximize the total reward accumulated in T rounds. The oracle's strategy with the knowledge of the fully observed action vectors and θ_* at each round t is $X_t^* = \arg\max_{X \in D_t} X^T \theta_*$. We evaluate the agent's performance against the oracle performance using *regret* defined as the difference between expected reward between the oracle and the agent,

$$R_T := \sum_{t=1}^{T} X_t^{*T} \theta_* - \sum_{t=1}^{T} X_t^T \theta_* = \sum_{t=1}^{T} (X_t^* - X_t)^T \theta_*. \tag{4.4}$$

4.4 Proposed Appraoch: Partially Observable Linear Bandits

In this section, we outline the methodology of our proposed algorithm. The core idea is to leverage a subspace recovery technique to estimate the underlying low-dimensional subspace from the partially observed action vectors at each round. Specifically, the algorithm first constructs a high-probability confidence set for the projection matrix, ensuring it contains the true projection matrix with high probability. In order to form this confidence set, we first estimate the underlying low-dimensional subspace. We elaborate on estimating the underlying sub-space in Section 4.4.1.

Using this subspace estimate, the algorithm imputes the missing entries in the partially observed action vectors at each round. This imputation process generates a high-probability confidence set for the complete action vectors, encapsulating the true action vectors. Subsequently, building on standard stochastic linear bandit (SLB) analysis, we derive the confidence set for the underlying model parameters.

With these three confidence sets—the projection matrix, the imputed action vectors, and the model parameters—our proposed approach employs the Optimism in the Face of Uncertainty (OFU) principle. We name our proposed algorithm Partially Observalble Linear Bandits (POLB). The algorithm optimistically selects an action that maximizes the reward based on the constructed confidence sets, effectively balancing exploration and exploitation. The algorithm is outlined in Algorithm 4. We briefly describe the 4 key components of : confidence set construction for projection matrix, confidence set construction for action vectors, confidence set construction for model parameter and acting optimistically based on OFU principle.

Algorithm 4 Partially Observable Linear bandits (POLB)

Inputs: $p, m, d, K, \dot{X}_{t,i} \quad \forall t \in [T]$ and $i \in [K]$
for $t = 1, \ldots, T$ **do**
    Receive a decision set D_t of partially observed action vectors $\dot{X}_{t,i}$ for $i \in [K]$ and estimate the subspace $\hat{\mathbf{U}}_t$ by performing PCA on covariance estimate in Equation (4.6).
    for $t > t_b$ **do**
        Compute the projection matrix: $\hat{\mathbf{P}}_t = \hat{\mathbf{U}}_t \hat{\mathbf{U}}_t^\top$
        Impute the entries of the partially observable action vectors using the operator $\mathcal{I}_t$ defined in Equation (4.9) and form the imputed decision set $\hat{D}_t$.
        Construct $\mathcal{C}_{\mathbf{P},t}$ given by Equation (4.14): high probability confidence set centered at $\hat{\mathbf{P}}_t$ which contains the true projection matrix.
        Construct $\mathcal{C}_{m,t}$ given by Equation (4.16), high probability confidence set centered at $\hat{\theta}_t$ which contains the true θ^*.
        Optimistically choose $(\tilde{\mathbf{P}}, \tilde{X}, \tilde{\theta}) = \arg\max_{(\mathbf{P},X,\theta) \in \mathcal{C}_{\mathbf{P},t} \times \mathcal{C}_{X,t} \times \mathcal{C}_{m,t}} (\mathbf{P}X)^\top \theta$
        Play $\hat{X}_t$ and observe r_t corresponding the true action vector X_t.
    end for
end for

4.4.1 Confidence Set Construction for Projection Matrix

Estimating the underlying subspace is a crucial preliminary step before constructing the confidence set for projection matrix. Given that the action vectors lie in an unknown m-dimensional subspace, the first step of POLB is to accurately estimate this subspace.

At each round an action vector set D_t is revealed whose cardinality is K. Each of the action vectors is partially observable based on the observability model given in Equation 4.2. Consider when the action vectors X_t are fully observable then we can compute the sample covariance matrix after T rounds as:

$$\Sigma_T = \frac{1}{t} \sum_{t=1}^{T} X_t X_t^\top. \tag{4.5}$$

The introduction of partially observable action vectors $\dot{X}_t$ renders the covariance

estimator in Equation (4.5) biased. Subsequently, performing principle component analysis (PCA) on this biased covariance estimator can lead to inaccurate subspace estimates. Therefore we use an unbiased estimator of the sample covariance matrix Σ_t given by,

$$\dot{\Sigma}_T = p^{-2}\Sigma_T^{(p)} + (p^{-1} - p^{-2})\text{diag}\left(\Sigma_T^{(p)}\right), \tag{4.6}$$

where, $\Sigma_T^{(p)} = \frac{1}{T}\sum_{t\in[T]} \dot{X}_i \dot{X}_i^\top$. It can be shown that this estimator is unbiased and we defer its proof to the appendix.

Performing PCA on this unbiased estimate of covariance matrix $\dot{\Sigma}_t$ we obtain an estimate of the subspace $\hat{\mathbf{U}}_t$. At each round the algorithm estimates the subspace $\hat{\mathbf{U}}_t$ based on the partially observed action vectors and we compute the projection matrix $\hat{\mathbf{P}}_t := \hat{\mathbf{U}}_t\hat{\mathbf{U}}_t^\top$. As the algorithm proceeds to collect more samples, the subspace estimation becomes more accurate and as a consequence so does the projection matrix estimate. As we do not know the true projection matrix $\mathbf{P}$, we construct a confidence set $\mathcal{C}_{\mathbf{P},t} = \{\|\hat{\mathbf{P}}_t - \mathbf{P}\|_2 \leq \epsilon_t\}$ around the estimated projection matrix which will contain the true projection matrix. Here, ϵ_t is an upper bound on the size of the ellipsoid which depends on the properties of the subspace and the degree of missing entries in the action vectors. As we collect more samples we will show that the size of this confidence set shrinks. This is analyzed in detail in Lemma 4 in Section 4.5.1.

4.4.2 Confidence Set Construction for Action Vectors

In this section, we aim to construct a confidence set that, with high probability, contains the true action vectors observed at each round. However, to build such a set, we first need to address the missing entries in the action vectors. This process involves imputing the missing values using the observed entries and the estimated subspace.

At each round t and for each index $i \in [K]$, the action vector $X_{t,i}$ is partially observed. The partially observed action vector is denoted as $\dot{X}_{t,i}$. We denote the set of observed indices as $\Omega_{t,i}$ i.e., i.e., $\dot{X}_{t,i}^{(\Omega_{t,i})} = X_{t,i}^{(\Omega_{t,i})}$ and $\dot{X}_{t,i}^{(\Omega_{t,i}^c)} = 0$. For ease of notation, we drop the the index i for subsequent analysis.

To impute the missing entries, we project the observed part of the action vector onto an estimated subspace spanned by the matrix $\hat{\mathbf{U}}$. Specifically, we solve the following least-squares problem:

$$\hat{\alpha} = \min_{\alpha \in \mathbb{R}^m} \|X_t^{(\Omega_t)} - \hat{\mathbf{U}}_{\Omega_t} \alpha\|_2^2. \tag{4.7}$$

where $\hat{\mathbf{U}}_{\Omega_t}$ is a sub-matrix of $\hat{\mathbf{U}}$ containing only the rows corresponding to the observed index set Ω_t. Then the solution to the above problem is given by:

$$\hat{\alpha} = (\hat{\mathbf{U}}_{\Omega_t}^\top \hat{\mathbf{U}}_{\Omega_t})^{-1} \hat{\mathbf{U}}_{\Omega_t}^\top X_t^{(\Omega_t)}.$$

The imputed values for the missing indices Ω_t^c are given by:

$$\hat{X}_t^{(\Omega_t^c)} = \hat{\mathbf{U}}_{\Omega_t^c} \hat{\alpha} = \hat{\mathbf{U}}_{\Omega_t^c} (\hat{\mathbf{U}}_{\Omega_t}^\top \hat{\mathbf{U}}_{\Omega_t})^{-1} \hat{\mathbf{U}}_{\Omega_t}^\top X_t^{(\Omega_t)} \tag{4.8}$$

The complete imputed action vector can then be expressed as: $\hat{X}_t = \mathcal{I}_t X_t$,

$$\mathcal{I}_t = \mathbf{I}_{\Omega_t}^\top \mathbf{I}_{\Omega_t} + \mathbf{I}_{\Omega_t^c}^\top \hat{\mathbf{U}}_{\Omega_t^c} (\hat{\mathbf{U}}_{\Omega_t}^\top \hat{\mathbf{U}}_{\Omega_t})^{-1} \hat{\mathbf{U}}_{\Omega_t}^\top \mathbf{I}_{\Omega_t} \tag{4.9}$$

where, $\mathcal{I}_t$ is the linear imputation operator.

Given the knowledge of true subspace $\mathbf{U}$ the true linear operator that imputes entries is:

$$\mathcal{I}(\Omega_t) = \mathbf{I}_{\Omega_t}^\top \mathbf{I}_{\Omega_t} + \mathbf{I}_{\Omega_t^c}^\top \mathbf{U}_{\Omega_t^c} (\mathbf{U}_{\Omega_t}^\top \mathbf{U}_{\Omega_t})^{-1} \mathbf{U}_{\Omega_t}^\top \mathbf{I}_{\Omega_t}$$

The imputation error vector at time t for an action vector X_t is:

$$E_t = X_t - \hat{X}_t = (\mathcal{I}(\Omega_t) - \mathcal{I}_t) X_t.$$

We aim to bound the 2-norm of E_t, as this will yield a confidence set where the true action vectors lie in.

4.4.3 Confidence Set Construction for Model Parameter

At each iteration, POLB aims to build a confidence set for the model parameter θ^*. We leverage the m-dimensional hidden subspace structure by constructing a confidence set, $\mathcal{C}_{m,t}$ around the estimated model parameter with a high probability. Utilizing the historical data of action-reward pairs, the algorithm solves a regularized least squares problem within the estimated subspace, yielding $\hat{\theta}_t$, the estimated model parameter vector within the span of $\hat{\mathbf{U}}_t$. Denoting the rewards obtained till round t is denoted as $\mathbf{r}_{t-1}$. The model parameter estimate $\hat{\theta}_t$ is the solution to the following problem:

$$\hat{\theta}_t = \arg\min_{\theta} \|(\hat{\mathbf{P}}_t \hat{\mathbf{X}}_{t-1})^\top \theta - \mathbf{r}_{t-1}\|_2^2 + \lambda \|\hat{\mathbf{P}}_t \theta\|_2^2, \tag{4.10}$$

where, λ is a regularization parameter and $\hat{\mathbf{X}}_t$ is a $d \times t$ matrix whose each column is given by the imputed action vector chosen by the algorithm at each round t. Subsequently, we construct the confidence set $\mathcal{C}_{m,t}$ around $\hat{\theta}_t$, ensuring that θ^* is highly probable to fall within $\mathcal{C}_{m,t}$. This is discussed in detail in Section 4.5.3

4.4.4 Choosing an Optimistic Action

The final step of our algorithm is to choose an optimistic tuple $\tilde{\mathbf{P}}_t, \tilde{X}_t, \tilde{\theta}_t$, from the confidence sets created $\mathcal{C}_{\mathbf{P},t}, \mathcal{C}_{m,t}$ and $\mathcal{C}_{X_t}$. We do this by solving the following optimization problem:

$$(\tilde{\mathbf{P}}, \tilde{X}, \tilde{\theta}) = \underset{\substack{\mathbf{P}, X, \theta \in \\ \mathcal{C}_{\mathbf{P},t} \times \mathcal{C}_{X,t} \times \mathcal{C}_{m,t}}}{\operatorname{argmax}} \quad (\mathbf{P}X)^T \theta, \tag{4.11}$$

which jointly maximizes the reward.

4.5 Theoretical analysis of POLB

In this section we state the upper the bound on regret of POLB which is the main result of our paper. Before we derive the regret bound, there two important components that build up to the result : the projection error analysis (see, Section 4.5.1) and the construction of projected confidence set (see, Section 4.5.3).

4.5.1 Confidence Set Construction Analysis for Projection Matrix

For orthonormal basis matrices $\hat{\mathbf{U}}_t, \mathbf{U}$, we define $\text{dist}(\hat{\mathbf{U}}_t, \mathbf{U}) := \|(\mathbf{I} - \hat{\mathbf{U}}_t\hat{\mathbf{U}}_t^\top)\mathbf{U}\|_2$ as a measure of subspace error (distance) between them. This is equal to the sine of the largest principal angle between the subspaces. We state the following theoretical result based on the estimated subspace:

Lemma 3 *With probability at least $1-\delta$ the following holds for all $t \in [T]$*

$$dist(\hat{\mathbf{U}}_t, \mathbf{U}) \leq \epsilon_t,$$

where,

$$\epsilon_t := \sqrt{\frac{12\lambda_1^2 mM^2}{\lambda_m^2 p^2 tK} \log\left(\frac{2dT}{\delta}\right)}. \tag{4.12}$$

Denote the event where Lemma 7 holds as $\mathcal{E}_1(\delta, t)$. The proof is provided in Appendix 4.11. Using the analysis from Akhiezer et al. 2013 it can be shown that:

$$\begin{aligned}\|\hat{\mathbf{P}}_t - \mathbf{P}\|_2 &= \sqrt{\lambda_{\max}\left(I_m - (\hat{\mathbf{U}}_t^\top \mathbf{U})^T, (\hat{\mathbf{U}}_t^\top \mathbf{U})\right)} \\ &= \sqrt{1 - \sigma^2(\hat{\mathbf{U}}_t^\top \mathbf{U})} = \sin\theta_1 \leq \epsilon_t. \end{aligned} \tag{4.13}$$

Lemma 4 *If $\mathcal{E}_1(\delta, t)$ occurs, with probability at least $1-\delta$, $\forall t \in [t_b, T]$, the following holds:*

$$\|\hat{\mathbf{P}}_t - \mathbf{P}\|_2 \leq \epsilon_t. \tag{4.14}$$

Lemma 4 also brings important intuitions about the subspace estimation problem in terms of the problem structure. As the subspace estimation problem becomes more challenging (i.e., as the probability of observation p decreases), the number of samples required for accurate estimation increases. This can be mitigated by observing more action vectors, K, at each time step. Similarly, if the condition number $\kappa = \frac{\lambda_1}{\lambda_m}$ increases, the confidence set of the projection matrix becomes looser. Likewise, as the subspace dimension m increases, the projection matrix confidence set also becomes looser. The effects of the structural properties of the subspace and the probability of observation p on the projection matrix confidence set directly influence the construction of confidence sets for the action vectors and model parameters, as discussed in the following sections. Ultimately, these factors contribute to the regret upper bound.

4.5.2 Confidence Set Construction Analysis for Action Vectors

To ensure accurate imputation of the missing entries, we must first learn the subspace with sufficient precision. This leads to the definition of a burn-in period, denoted as t_b, during which a sufficient number of samples is collected to guarantee that the distance measure, ϵ_t, is less than $\mathcal{O}(p)$. Since only a fraction p of the data is visible, the accuracy of the subspace estimation is fundamentally constrained by the missing data. Achieving an error of $\mathcal{O}(p)$ ensures that the subspace estimate is sufficiently close to the true subspace. To be specific, we set $\epsilon_t \leq p/32$ in Equation (4.12) to we derive the following condition for t_b:

$$t \geq \frac{C\lambda_1^2 m M^2}{\lambda_m^2 p^3 K} \log\left(\frac{2dT}{\delta}\right) = t_b, \tag{4.15}$$

where, C is some positive scalar. Based on the preceding analysis, we now present a confidence set that, with high probability, contains the true action vectors observed at each round.

Lemma 5 *With probability at least* $1 - 3\delta$ *and for all* $t \in [t_b, T]$*, the true action vectors* $\mathbf{X}_t = \{X_{t,1}, \ldots, X_{t,K}\} \in \mathbb{R}^{d \times K}$ *lie in the following confidence set:*

$$\mathbf{X}_t \in \mathcal{C}_{X,t} = \prod_{i=1}^{K} \mathcal{C}_{X_{t,i}},$$

where, each $X_{t,i}$ *satisfies:*

$$X_{t,i} \in \mathcal{C}_{X_{t,i}} = \left\{ X \in \mathbb{R}^d : \|\hat{X}_{t,i} - X\|_2 \leq \left(1 + \frac{2}{p}\right) L\epsilon_t \right\}.$$

Here, $\hat{X}_{t,i}$ *is the* $i^{th}-$*imputed action vector at time* t *computed using Equation* (4.8)*, and the parameter* p *satisfies the following condition:*

$$p \geq \min\left\{ 16\frac{\mu^2 m}{d} \log\left(\frac{mTK}{\delta}\right), 1 \right\},$$

where μ *is the coherence parameter and* ϵ_t *is the subspace estimation error given in Equation* (4.12)*.*

The detailed proof is deferred to Section 4.12 of the Appendix. We denote the event where Lemma 5 holds as $\mathcal{E}_2(\delta, t)$. Recall the definition of ϵ_t from Equation (4.12). Each of the confidence sets $\mathcal{C}_{X_{t,i}}$ is reliant on the structural properties of the underlying subspace problem as well as the number of samples tK. Therefore, as we observe an increasing number of samples, the confidence set shrinks, and the imputed action vectors converge more closely to the true action vectors.

4.5.3 Confidence Set Construction Analysis for Model Parameter

In this section we analyze the construction of projected confidence set $\mathcal{C}_{m,t}$. Recalling the least squares problem defined in Equation (4.10). Solving for θ gives us an estimate

$\hat{\theta}_t = A_t^{\dagger}(\hat{\mathbf{P}}_t \hat{\mathbf{X}}_{t-1} \mathbf{r}_{t-1})$. Here, $A_t = \hat{\mathbf{P}}_t \left(\hat{\Sigma}_{t-1} + \lambda I_d\right) \hat{\mathbf{P}}_t$. Note that, $\hat{\Sigma}_t = \sum_t \hat{X}_t \hat{X}_t^{\top} = \hat{\mathbf{X}}_t \hat{\mathbf{X}}_t^{\top}$, is formed by the imputed action vectors. The following theorem gives the construction of projected confidence set, $\mathcal{C}_{m,t}$, which is an ellipsoid centered around θ_t which contains θ_* with high probability.

Theorem 4 *Let the action vectors $\|X_t\|_2 \leq L$ and $\|\theta_*\|_2 \leq S$, then for all $t \geq t_b$ with probability at least $1 - 4\delta$, θ_* lies in the set*

$$\mathcal{C}_{m,t} = \left\{\theta \in \mathbb{R}^d : \|\hat{\theta}_t - \theta\|_{A_t} \leq \beta_{t,\delta}\right\},$$

where,

$$\beta_{t,\delta} := R\sqrt{2\log\left(\frac{T}{\delta}\right) + 8m\log\mathcal{L}} + 8L\left(1 + \left(1 + \frac{2}{p}\right)\sqrt{\frac{\zeta(1+\log t)}{t\lambda}}\right)\sqrt{m\zeta\log\mathcal{L}} + 8SL\left(1 + \frac{2}{p}\right)\sqrt{m\zeta(1+\log t)\log\mathcal{L}} + S\sqrt{\lambda}. \tag{4.16}$$

Here, $\mathcal{L} = \left(1 + \frac{\zeta L^2(1+\log t)}{m\lambda}\left(1 + \frac{2}{p}\right)^2 + \frac{tL^2}{m\lambda}\right)$, $\zeta = \frac{12m\lambda_1^2 M^2}{p^2\lambda_m^2 K}\log\left(\frac{2dT}{\delta}\right)$ *and* $p \geq \min\left\{16\frac{\mu^2 m}{d}\log\left(\frac{mTK}{\delta}\right), 1\right\}$.

The detailed proof is deferred to Section 4.13 of the Appendix. At first glance, we can see some similarities between the size of the ellipsoid we construct and the one in Yasin Abbasi-Yadkori et al. 2011. However, since POLB projects the imputed action set onto the estimated m-dimensional subspace, the ambient dimension d is replaced by m in the size of the ellipsoid. While improves on the previous bound by utilizing the underlying low-dimensional structure of the problem, this construction of the confidence set suffers from finite sample projection error and error in imputing the missing entries. This is seen in the second and third terms of Equation (4.16).

Given that we have now constructed the confidence set $\mathcal{C}_{m,t}$, we search for the optimistic model parameter θ_*. Optimistically choosing the tuple $\tilde{\mathbf{P}}_t, \tilde{X}_t, \tilde{\theta}_t$, is given in Equation (4.11).

4.5.4 Regret Analysis

In this section, we present the regret analysis of using the projected confidence set $\mathcal{C}_{m,t}$ and elaborate further on the effect of subspace recovery difficulty based on the structural properties of the problem. The detailed proof analysis follows similar methods as in Yasin Abbasi-Yadkori et al. 2011; Lale et al. 2019, which we defer to Appendix 4.14.

Theorem 5 (Main Result) *For $T \geq t_b$, the following regret bound holds with probability $1 - 4\delta$:*

$$R_T \in \tilde{\mathcal{O}}\left(\frac{m^{3/2}\lambda_1^2\sqrt{T}}{p^3\lambda_m^2 K}\right) \tag{4.17}$$

with probability of observation p satisfying the following condition:

$$p \geq \min\left\{16\frac{\mu^2 m}{d}\log\left(\frac{mTK}{\delta}\right), 1\right\}.$$

The following remarks summarize key aspects of our method's regret bound:

1. **Dependence on condition number**: The regret bound is influenced by the condition number of the subspace, λ_1^2/λ_m^2. A larger condition number indicates that the subspace is ill-conditioned, making it more difficult to recover accurately and leading to higher regret. On the other hand, when the condition number is small, the subspace recovery becomes more stable, reducing the regret.

2. **Regret scaling with observation probability** p: When the probability of observation p is constant, i.e., $p = \mathcal{O}(1)$, our method achieves a regret bound of $\tilde{\mathcal{O}}(m^{3/2}\sqrt{T})$. This result is significantly better than the standard linear bandit regret bound of $\tilde{\mathcal{O}}(d\sqrt{T})$, especially in cases where $m \ll d$.

Another interesting regime is when $p \sim \frac{m^{1/2}}{f(d)}$. In this case, the regret R_T scales like $f(d)^3\sqrt{T}$, which is significantly better than linear bandits in situations where $f(d) \ll d^{1/3}$ and $m \leq d^{2/3}$.

For instance, if $m \sim \log d$, $p \sim 1/\log d$, then regret $R_T \lesssim \text{polylog}(d)\sqrt{T}$ which is exponentially better than linear bandits with a vanishly small fraction of the action/context vectors observed.

4.6 Experimental Studies

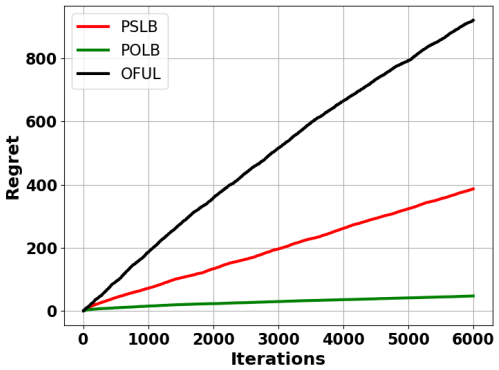
We evaluate our proposed method, POLB , against established algorithms such as Optimism in the Face of Uncertainty (OFUL) Yasin Abbasi-Yadkori et al. 2011 and Projected Stochastic Linear Bandits (PSLB) Lale et al. 2019. It is important to note that OFUL does not leverage the underlying hidden low-rank structure in the data. While PSLB is designed to exploit this hidden low-rank structure, it assumes access to the complete action vectors. In contrast, our experimental setup involves observing only partial action vectors, making PSLB unsuitable for handling the missing observations effectively.

4.6.1 Synthetic Experiment

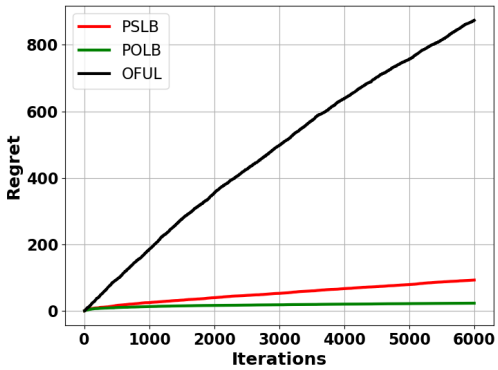
We consider the case where the ambient dimension is $d = 100$, and the underlying subspace has dimension $m = 2$. The eigenvalues of the subspace were chosen to be $\{1, 0.5\}$. The block size B is 20. The different values of the probability of observation we consider are $\{0.2, 0.5, 0.7\}$. achieves the lowest regret across different observation probability regimes. This is demonstrated in Figure 17.

4.6.2 Image Classification

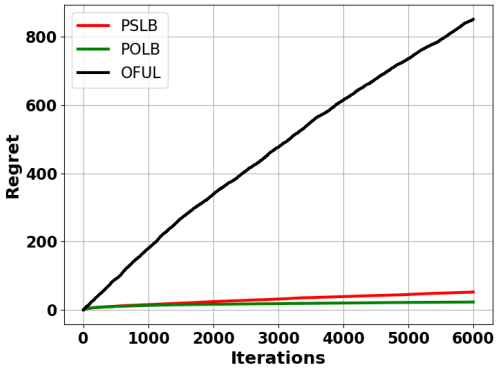
In this experiment, we study the MNIST dataset and use it to create decision sets for the SLB setting. A simple 5-layer CNN is deployed for MNIST. Before training, we modify the architecture of the representation layer (the layer before the final output



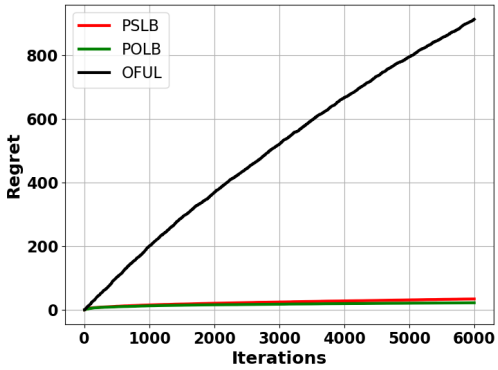
(a) Observation probability $p = 0.2$.



(b) Observation probability $p = 0.5$.



(c) Observation probability $p = 0.7$.



(d) Observation probability $p = 0.9$.

Figure 17. Regret comparisons of POLB, PSLB, and OFUL on synthetic dataset with ambient dimension $d = 100$, $m = 2$.

layer) to make it suitable for the SLB study and generate decision sets for each image. Consider a standard network where the dimension of the representation layer is d. In this network, the final layer for K-class classification is fully connected and is represented by a $d \times K$ matrix, which outputs K values for classification.

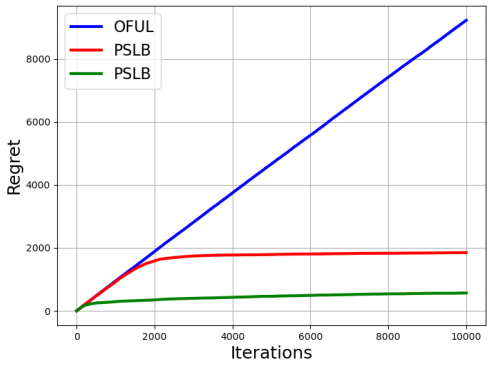
In this study, instead of using a final layer of size $d \times K$, we construct the final layer as a d-dimensional vector and make the feature representation layer a Kd-dimensional vector. This vector is treated as the concatenation of K d-dimensional contexts, i.e., $[X_1, \ldots, X_K]$. The final d-dimensional layer, denoted as θ_*, represents the SLB

parameters, where the logit for each class is computed as the inner product of the class context X_i and θ_*. These architectures are trained for different values of d using cross-entropy loss.

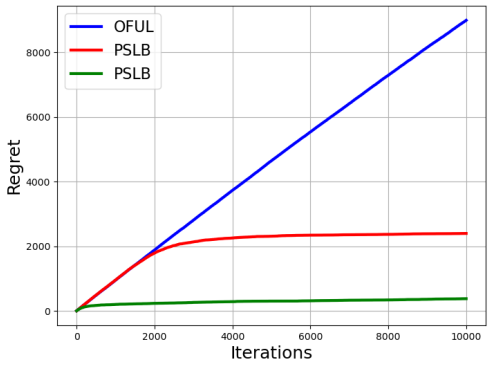
After removing the final layer, the resulting trained networks are used to generate the feature representations of each image for each class, producing the decision sets at each time step of the SLB. Since MNIST consists of 10 classes, each decision set contains 10 action vectors, with each vector representing a segment of the representation layer. In the SLB setting, the agent receives a reward of 1 if it selects the correct action (the segment in the representation layer corresponding to the correct label according to the trained network) and 0 otherwise.

To simulate partial observability, we sparsify the entries in the Kd-dimensional representation layer based on a given observation probability p. This introduces missing values in the representation, reflecting the probability p of observing each entry. Here, we present results for MNIST with $d = 300$ for different values of p.

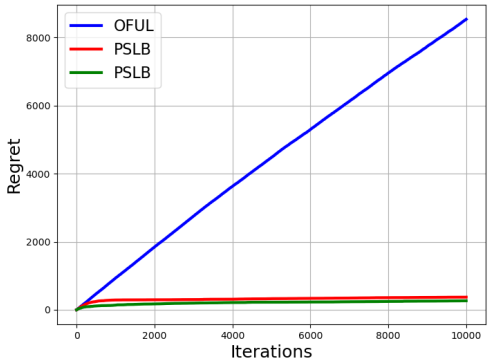
During the experiments, POLB attempted to recover an $8-$dimensional subspace using the partially observed action vectors collected. We present the regrets obtained by PSLB and OFUL for MNIST in Figure 18. With the aid of subspace recovery and projection, POLB significantly reduces the dimensionality of the SLB problem and quickly estimates an accurate model for θ_*. In contrast, OFUL naively attempts to sample from all dimensions to learn θ_*, leading to orders of magnitude worse performance in terms of regret. Although PSLB utilizes subspace recovery and projection, it is not designed to handle partially observed action vectors, which limits its performance compared to our POLB. However, as the probability of observing entries in the action vectors increases, PSLB's regret performance approaches that of POLB.



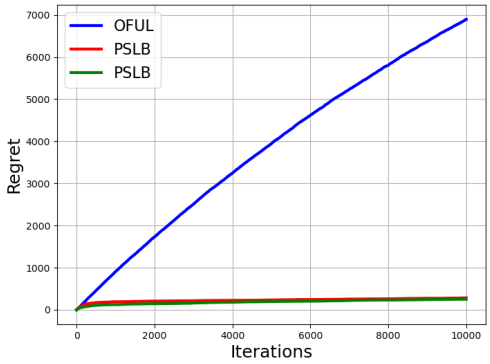
(a) Observation probability $p = 0.2$.



(b) Observation probability $p = 0.5$.



(c) Observation probability $p = 0.7$.



(d) Observation probability $p = 0.9$.

Figure 18. Regret comparisons of POLB, PSLB, and OFUL on MNIST dataset with representation layer dimension $d = 300$, $K = 10$, $m = 8$.

4.7 Conclusion

In this work, a novel algorithm POLB for stochastic linear bandits with partially observed action vectors. Our approach leverages subspace recovery techniques to estimate the underlying low-dimensional structure of the action space, even when only partial observations are available. By constructing confidence sets for the projection matrix, imputed action vectors, and model parameters, POLB employs a triple optimism strategy to effectively balance exploration and exploitation. The key

contributions of this work include a framework for handling partially observed action vectors in stochastic linear bandits, addressing an important challenge in modern sequential decision-making applications. We propose an algorithm that adapts to the inherent low-dimensional structure of the action space, enabling efficient learning even in high-dimensional settings. Our theoretical analysis demonstrates that POLB achieves regret bounds that depend on the subspace dimension m and observation probability p, rather than the ambient dimension d. Additionally, we provide insights into the trade-offs between subspace dimension, observation probability, and regret performance.

Our results highlight the potential for leveraging low-dimensional structures in partially observed settings to improve the efficiency and scalability of bandit algorithms. This work opens up several promising directions for future research, such as extending the framework to handle time-varying subspaces or non-linear reward models, investigating the impact of different missing data patterns or observation models, and developing adaptive strategies for estimating the subspace dimension m in online settings. Furthermore, the approach can be applied to domains such as recommendation systems, online advertising, and clinical trials, where partial observability is common. By addressing the challenges of high-dimensionality and partial observability, POLB represents a significant step toward more practical and robust bandit algorithms for real-world applications. As sequential decision-making problems continue to grow in complexity and scale, techniques that can efficiently handle limited or incomplete information will become increasingly valuable.

4.8 Proof of Theorems and Lemmas

4.9 Unbiased Estimator of Covariance Matrix with Missing Entries

Lemma 6 *Let $\dot{X}_t$ be the partially observed action vector according to the generative model in Equation 4.1 and observability model in Equation 4.2, then the estimator $\dot{\Sigma}_T = p^{-2}\Sigma_T^{(p)} + (p^{-1} - p^{-2})diag(\Sigma_T^{(p)})$, where, $\Sigma_T^{(p)} = \frac{1}{T}\sum_{t\in[T]} \dot{X}_i\dot{X}_i^\top$ is an unbiased estimator of the covariance matrix Σ.*

Proof: If the action vector X_t is fully observed, we can compute the sample covariance matrix after T rounds as,

$$\Sigma_T = \frac{1}{T}\sum_{t=1}^{T} X_T X_T^\top. \tag{4.18}$$

The introduction of partially observable action vectors $\dot{X}_t$ according to the generative model in Equations (4.1) and (4.2) renders the covariance estimator in Equation (4.18) biased. An unbiased estimator of the sample covariance matrix Σ_T is then given by,

$$\dot{\Sigma}_T = p^{-2}\Sigma_T^{(p)} + (p^{-1} - p^{-2})\text{diag}(\Sigma_T^{(p)}), \tag{4.19}$$

where, $\Sigma_T^{(p)} = \frac{1}{T}\sum_t \dot{X}_t\dot{X}_t^\top$. We now define a quantity,

$$\Sigma^{(p)} = (p - p^2)\text{diag}(\Sigma_T) + p^2\Sigma_T. \tag{4.20}$$

It is easy to check that $\mathbb{E}[\Sigma_T^{(p)}] = \Sigma^{(p)}$. By simple observation we can see that,

$$\Sigma_T = (p^{-1} - p^{-2})\text{diag}(\Sigma^{(p)}) + p^{-2}\Sigma^{(p)}. \tag{4.21}$$

Therefore, taking expectation on both sides of (4.19), we see that $\dot{\Sigma}_t$ is an unbiased estimator of sample covariance matrix.

■

Lemma 7 *Let* $\dot{\Sigma}_n = \frac{1}{p^2 n}\sum_{t\in[n]} \dot{X}_t\dot{X}_t^\top + \frac{1}{n}\left(\frac{1}{p} - \frac{1}{p^2}\right)\sum_{t\in[n]} diag\left(\dot{X}_t\dot{X}_t^\top\right)$, *where* $\dot{X}_t$*'s are the partially observed action vectors. Also,* $\Sigma = \mathbf{U}\Lambda^2\mathbf{U}^\top + \sigma^2 I$. *With probability at least* $1-\delta$*:*

$$\left\|\dot{\Sigma}_n - \Sigma\right\|_2 \leq \epsilon, \tag{4.22}$$

where,

$$\epsilon \leq \sqrt{\frac{12\lambda_1^2 m M^2}{p^2 n}\log\left(\frac{2d}{\delta}\right)}. \tag{4.23}$$

Proof: Define

$$\dot{\Sigma}_n = \frac{1}{p^2 n}\sum_{t\in[n]} \dot{X}_t\dot{X}_t^\top + \frac{1}{n}\left(\frac{1}{p} - \frac{1}{p^2}\right)\sum_{t\in[n]} \text{diag}\left(\dot{X}_t\dot{X}_t^\top\right).$$

Zero mean, independent quantity :

$$Q_t = \left(\frac{1}{p^2}\dot{X}_t\dot{X}_t^\top + \left(\frac{1}{p} - \frac{1}{p^2}\right)\text{diag}\left(\dot{X}_t\dot{X}_t^\top\right) - \Sigma\right)$$

We want to compute the variance of the above zero mean quantity:

$$\left\|\sum_t \mathbb{E}\left[(1/n)Q_t^2\right]\right\|_2 = 1/n\left\|\sum_t \mathbb{E}\left[Q_t^2\right]\right\|_2$$

Expanding on the expected value of Q_t^2:

$$\begin{aligned}
\mathbb{E}\left[Q_t^2\right] = \mathbb{E}\Bigg(&\frac{1}{p^2}\dot{X}_t\dot{X}_t^\top + \left(\frac{1}{p} - \frac{1}{p^2}\right)\operatorname{diag}\left(\dot{X}_t\dot{X}_t^\top\right) - \Sigma\Bigg) \\
&\times \left(\frac{1}{p^2}\dot{X}_t\dot{X}_t^\top + \left(\frac{1}{p} - \frac{1}{p^2}\right)\operatorname{diag}\left(\dot{X}_t\dot{X}_t^\top\right) - \Sigma\right)^\top \\
= \mathbb{E}\Bigg(&\frac{1}{p^4}\dot{X}_t\dot{X}_t^\top\dot{X}_t\dot{X}_t^\top + \frac{1}{p^2}\left(\frac{1}{p} - \frac{1}{p^2}\right)\dot{X}_t\dot{X}_t^\top\operatorname{diag}\left(\dot{X}_t\dot{X}_t^\top\right)^\top \\
&- \frac{1}{p^2}\dot{X}_t\dot{X}_t^\top\Sigma^\top + \left(\frac{1}{p} - \frac{1}{p^2}\right)\operatorname{diag}\left(\dot{X}_t\dot{X}_t^\top\right)\frac{1}{p^2}\dot{X}_t\dot{X}_t^\top \\
&+ \left(\frac{1}{p} - \frac{1}{p^2}\right)^2\operatorname{diag}\left(\dot{X}_t\dot{X}_t^\top\right)\operatorname{diag}\left(\dot{X}_t\dot{X}_t^\top\right)^\top \\
&- \left(\frac{1}{p} - \frac{1}{p^2}\right)\operatorname{diag}\left(\dot{X}_t\dot{X}_t^\top\right)\Sigma^\top \\
&- \Sigma\frac{1}{p^2}\dot{X}_t\dot{X}_t^\top - \Sigma\left(\frac{1}{p} - \frac{1}{p^2}\right)\operatorname{diag}\left(\dot{X}_t\dot{X}_t^\top\right)^\top + \Sigma\Sigma^\top\Bigg) \\
= \mathbb{E}\Bigg(&\frac{1}{p^4}\|\dot{X}_t\|^2\dot{X}_t\dot{X}_t^\top + \frac{1}{p^2}\left(\frac{1}{p} - \frac{1}{p^2}\right)\dot{X}_t\dot{X}_t^\top\operatorname{diag}\left(\dot{X}_t\dot{X}_t^\top\right) \\
&- \frac{1}{p^2}\dot{X}_t\dot{X}_t^\top\Sigma^\top + \left(\frac{1}{p} - \frac{1}{p^2}\right)\operatorname{diag}\left(\dot{X}_t\dot{X}_t^\top\right)\frac{1}{p^2}\dot{X}_t\dot{X}_t^\top \\
&+ \left(\frac{1}{p} - \frac{1}{p^2}\right)^2\operatorname{diag}\left(\dot{X}_t\dot{X}_t^\top\right)^2 - \left(\frac{1}{p} - \frac{1}{p^2}\right)\operatorname{diag}\left(\dot{X}_t\dot{X}_t^\top\right)\Sigma^\top \\
&- \Sigma\frac{1}{p^2}\dot{X}_t\dot{X}_t^\top - \Sigma\left(\frac{1}{p} - \frac{1}{p^2}\right)\operatorname{diag}\left(\dot{X}_t\dot{X}_t^\top\right) + \Sigma\Sigma^\top\Bigg)
\end{aligned}$$

We know the following:

- $\|X\| \leq \|\mathbf{U}\|\|\Lambda\|\|Z_t\| \leq \sqrt{m}\lambda_1 M$
- $\mathbb{E}[\dot{X}_t\dot{X}_t^\top] = p^2\Sigma^N + p\Sigma^D$

Now we focus on bounding $\|\mathbb{E}\left[Q_t^2\right]\|_2$:

$$\begin{aligned}
\|\mathbb{E}Q_t^2\| &\leq \frac{1}{p^4}(\sqrt{m}\lambda_1 M)^2(p^2\Sigma^N + p\Sigma^D) - 2\frac{1}{p^2}(p^2\Sigma^N + p\Sigma^D)\Sigma \\
&\quad + 2\left(\frac{1}{p} - \frac{1}{p^2}\right)\frac{1}{p^2}\mathbb{E}\left[\operatorname{diag}\left(\dot{X}_t\dot{X}_t^\top\right)\dot{X}_t\dot{X}_t^\top\right] \\
&\quad + p(\sqrt{m}\lambda_1 M)^2\left(\frac{1}{p} - \frac{1}{p^2}\right)^2\Sigma^D - 2p\left(\frac{1}{p} - \frac{1}{p^2}\right)\Sigma^D\Sigma + \Sigma^2 \\
&\leq \frac{1}{p^4}(\sqrt{m}\lambda_1 M)^2(p^2\Sigma^N + p\Sigma^D) - 2\frac{1}{p^2}(p^2\Sigma^N + p\Sigma^D)\Sigma \\
&\quad + 2\left(\frac{1}{p} - \frac{1}{p^2}\right)\frac{1}{p^2}(p\Sigma^D + p^2\Sigma^N)\|\dot{X}_t\|^2 \\
&\quad + p(\sqrt{m}\lambda_1 M)^2\left(\frac{1}{p} - \frac{1}{p^2}\right)^2\Sigma^D - 2p\left(\frac{1}{p} - \frac{1}{p^2}\right)\Sigma^D\Sigma + \Sigma^2 \\
&\leq \frac{1}{p^4}(\sqrt{m}\lambda_1 M)^2(p^2\Sigma^N + p\Sigma^D) - 2\frac{1}{p^2}(p^2\Sigma^N + p\Sigma^D)\Sigma \\
&\quad + 2(\sqrt{m}\lambda_1 M)^2\left(\frac{1}{p^3} - \frac{1}{p^4}\right)(p\Sigma^D + p^2\Sigma^N) \\
&\quad + (\sqrt{m}\lambda_1 M)^2(p-1)^2\Sigma^D - 2\left(1 - \frac{1}{p}\right)\Sigma^D\Sigma + \Sigma^2
\end{aligned}$$

Simplifying by cancelling terms:

$$\begin{aligned}
&= \frac{2}{p^3}(\sqrt{m}\lambda_1 M)^2(p^2\Sigma^N + p\Sigma^D) - \frac{1}{p^4}(\sqrt{m}\lambda_1 M)^2(p^2\Sigma^N + p\Sigma^D) - \Sigma^2 \\
&\quad + (\sqrt{m}\lambda_1 M)^2(p-1)^2\Sigma^D \\
&\leq \frac{2}{p^3}(\sqrt{m}\lambda_1 M)^2(p\Sigma^N + p\Sigma^D) - \frac{1}{p^4}(\sqrt{m}\lambda_1 M)^2(p^2\Sigma^N + p^2\Sigma^D) \\
&\quad + (\sqrt{m}\lambda_1 M)^2\Sigma^D \\
&= \frac{1}{p^2}(\sqrt{m}\lambda_1 M)^2\Sigma + (\sqrt{m}\lambda_1 M)^2\Sigma^D.
\end{aligned}$$

Computing

$$\begin{aligned}
&\left\|\frac{1}{n}\sum_t \mathbb{E}\left[\left(\frac{1}{p^2}\dot{X}_t\dot{X}_t^\top + \left(\frac{1}{p} - \frac{1}{p^2}\right)\operatorname{diag}\left(\dot{X}_t\dot{X}_t^\top\right) - \Sigma\right)\right.\right. \\
&\qquad \left.\left.\times\left(\frac{1}{p^2}\dot{X}_t\dot{X}_t^\top + \left(\frac{1}{p} - \frac{1}{p^2}\right)\operatorname{diag}\left(\dot{X}_t\dot{X}_t^\top\right) - \Sigma\right)^\top\right]\right\|
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{n}\left(\left\|\frac{1}{p^2}(\sqrt{m}\lambda_1 M)^2\Sigma + (\sqrt{m}\lambda_1 M)^2\Sigma^D\right\|\right) \\
&\leq \frac{(\sqrt{m}\lambda_1 M)^2}{n}\left(\frac{1}{p^2}+1\right) \\
&\leq \frac{2(\sqrt{m}\lambda_1 M)^2}{np^2}.
\end{aligned}
$$

Operator norm of zero mean quantity $\|Q_t\|_2$:

$$
\begin{aligned}
\|Q_t\| &= \left\|\frac{1}{p^2}\dot{X}_t\dot{X}_t^\top + \left(\frac{1}{p}-\frac{1}{p^2}\right)\operatorname{diag}\left(\dot{X}_t\dot{X}_t^\top\right) - \Sigma\right\| \\
&\leq \frac{1}{p^2}\|\dot{X}_t\|^2 + \left(\frac{1}{p}-\frac{1}{p^2}\right)\max_i\left|X_t^{(i)}\right|^2 + \|\dot{X}_t\|^2 \\
&= (\sqrt{m}\lambda_1 M)^2\left(\frac{1}{p^2}+1\right) + \left(\frac{1}{p}-\frac{1}{p^2}\right)\frac{(\sqrt{m}\lambda_1 M)^2}{d} \\
&\leq \frac{2(\sqrt{m}\lambda_1 M)^2}{p^2} + \frac{(\sqrt{m}\lambda_1 M)^2}{pd} \\
&\leq \frac{3(\sqrt{m}\lambda_1 M)^2}{p^2}.
\end{aligned}
$$

Using matrix Bernstein's inequality we can say that with probability at least $1-\delta$:

$$
\mathbb{P}\left(\left\|\dot{\Sigma}_n - \Sigma\right\| \leq \epsilon\right),
$$

where,

$$
\epsilon \leq \sqrt{\frac{12\lambda_1^2 m M^2}{p^2 n}\log\left(\frac{2d}{\delta}\right)}. \tag{4.24}
$$

■

4.11 Subspace Estimation Error (Proof of Lemma 3)

Before we state the proof of subspace estimation error, it is important to state the following theorem.

Theorem 6 (Davis & Kahan, 1970) *Let $S, \hat{S} \in \mathbb{R}^{d\times d}$ be symmetric matrices, such that $\hat{S} = S + H$. The eigenvalues of S and $\hat{S}$ are $\lambda_1 \geq \cdots \geq \lambda_m \geq \cdots \geq \lambda_d$ and $\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_m \geq \cdots \geq \hat{\lambda}_d$ respectively. Define the eigendecompositions of S and $\hat{S}$ as:*

$$S = \begin{bmatrix} \mathbf{U} & \mathbf{U}_o \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda} & 0 \\ 0 & \mathbf{\Lambda}_o \end{bmatrix} \begin{bmatrix} \mathbf{U}^T \\ \mathbf{U}_o^T \end{bmatrix},$$

$$\hat{S} = \begin{bmatrix} \hat{\mathbf{U}} & \hat{\mathbf{U}}_o \end{bmatrix} \begin{bmatrix} \hat{\mathbf{\Lambda}} & 0 \\ 0 & \hat{\mathbf{\Lambda}}_o \end{bmatrix} \begin{bmatrix} \hat{\mathbf{U}}^T \\ \hat{\mathbf{U}}_o^T \end{bmatrix},$$

where $\mathbf{\Lambda}$ *and* $\hat{\mathbf{\Lambda}}$ *are diagonal matrices with the first* m *eigenvalues of* S *and* $\hat{S}$ *respectively.* $\mathbf{U} = (u_1, \ldots, u_m) \in \mathbb{R}^{d \times m}$ *and* $\hat{\mathbf{U}} = (\hat{\mathbf{U}}_1, \ldots, \hat{\mathbf{U}}_m) \in \mathbb{R}^{d \times m}$ *denote the corresponding eigenvectors. Define*

$$\delta := \inf\{|\hat{\lambda} - \lambda| : \lambda \in [\lambda_m, \lambda_1], \hat{\lambda} \in (-\infty, \hat{\lambda}_{m+1}]\}.$$

If $\delta > 0$*, then* $\sin\theta_1$*, the sine of the largest principal angle between the column spans of* $\mathbf{U}$ *and* $\hat{\mathbf{U}}$*, can be upper bounded as*

$$\sin\theta_1 \leq \frac{\|\hat{S}\mathbf{U} - \mathbf{U}\mathbf{\Lambda}\|_2}{\delta} = \frac{\|\hat{S}\mathbf{U} - \mathbf{U}\mathbf{\Lambda}\|_2}{|\lambda_m - \hat{\lambda}_{m+1}|}.$$

Notice that in order to use the Davis-Kahan $\sin\Theta$ theorem in our setting, we need to pick two symmetric matrices S and $\hat{S}$ such that their first m eigenvectors have the same span as the subspaces that Σ and $\hat{\Sigma}$ span.

Using the above Davis-Kahan theorem and Lemma 7, we state the proof of Lemma 3.

Proof: By choosing $S = \Sigma$ and $\hat{S} = \dot{\Sigma}$, we get:

$$\text{dist}(\hat{\mathbf{U}}_t, \mathbf{U}) = \sin\theta_1 \overset{(a)}{\leq} \frac{\|(\hat{S} - S)\mathbf{U}\|_2}{\lambda_m(S) - \lambda_{m+1}(\hat{S})} \leq \frac{\|(\hat{S} - S)\|_2}{\lambda_m(S)} = \frac{\|\dot{\Sigma} - \Sigma\|_2}{\lambda_m(\Sigma)}. \tag{4.25}$$

Here, inequality (a) is a result of using Davis-Kahan $\sin\Theta$ theorem. By utilizing Lemma 7, we get the following bound which holds with probability $1-\delta$:

$$\text{dist}(\hat{\mathbf{U}}_t, \mathbf{U}) \leq \sqrt{\frac{12\lambda_1^2 mM^2}{p^2\lambda_m^2 tK}\log\left(\frac{2d}{\delta}\right)}. \tag{4.26}$$

This hold for any given time t. We want this to hold for all $t \in [T]$. Choosing probability of error as δ/T instead of just δ and computing union bound over T we can say that,

$$\text{dist}(\hat{\mathbf{U}}_t, \mathbf{U}) \leq \sqrt{\frac{12\lambda_1^2 mM^2}{p^2\lambda_m^2 tK}\log\left(\frac{2dT}{\delta}\right)}.$$

holds with probability $1-\delta$ for all $t \in [T]$.

■

4.12 Imputation Error (Proof of Lemma 5)

If we know the true underlying subspace $\mathbf{U}$ and the fully observed action vector $X_t \in \mathbb{R}^d$ then we know there exists a unique $\alpha \in \mathbb{R}^m$ such that: $X_t = \mathbf{U}\alpha$. Since, $\|X_t\|_2 \leq L$, we can say that $\|\mathbf{U}\alpha\|_2 \leq L$. At time t, given that we observe indices specified by Ω_t, the missing entries of true action vector can be written as:

$$X_t^{(\Omega_t^c)} = \mathbf{U}_{\Omega_t^c}\alpha,$$

Similarly, for the imputed action vector $\hat{X}_t$ we can the entries on the missing indices as:

$$\begin{aligned}\hat{X}_t^{(\Omega_t^c)} &= \hat{\mathbf{U}}_{\Omega_t^c}\hat{\alpha} \\ &= \hat{\mathbf{U}}_{\Omega_t^c}(\hat{\mathbf{U}}_{\Omega_t}^\top\hat{\mathbf{U}}_{\Omega_t})^{-1}\hat{\mathbf{U}}_{\Omega_t}^\top X_t^{(\Omega_t)}\end{aligned}$$

Next, we define an event where at least m entries are observed. This is to ensure that operator $\mathcal{I}_t$ is well defined, with the inverse term $\hat{\mathbf{U}}_{\Omega_t}^\top\hat{\mathbf{U}}_{\Omega_t}$ within it being invertible.

Observing at least m entries will make $\hat{\mathbf{U}}_{\Omega_t}^\top \hat{\mathbf{U}}_{\Omega_t}$ a full rank matrix. Lemma 9 specifies the value of p for which this happens.

Observing at least m entries will make the matrix $\hat{\mathbf{U}}_{\Omega_t}^\top \hat{\mathbf{U}}_{\Omega_t}$ invertible. However, to bound the imputation error, we needs its eigenvalue value to sufficiently away from zero. Given an index set at time t, Ω_t the following lemma states how well the minimum eigenvalue of $\hat{\mathbf{U}}_{\Omega_t}^\top \hat{\mathbf{U}}_{\Omega_t}$ behaves.

Imputation error at time t is:

$$
\begin{aligned}
\|E_t\|_2 &= \|X_t^{(\Omega_t^c)} - \hat{X}_t^{(\Omega_t^c)}\|_2 \\
&= \|\mathbf{U}_{\Omega_t^c}\alpha - (\hat{\mathbf{U}}_{\Omega_t^c}(\hat{\mathbf{U}}_{\Omega_t}^\top \hat{\mathbf{U}}_{\Omega_t})^{-1}\hat{\mathbf{U}}_{\Omega_t}^\top \mathbf{I}_{\Omega_t} X_t)\|_2 \\
&= \|\mathbf{U}_{\Omega_t^c}\alpha - (\hat{\mathbf{U}}_{\Omega_t^c}(\hat{\mathbf{U}}_{\Omega_t}^\top \hat{\mathbf{U}}_{\Omega_t})^{-1}\hat{\mathbf{U}}_{\Omega_t}^\top \mathbf{U}_{\Omega_t}\alpha)\|_2 \\
&\leq \|\mathbf{U}_{\Omega_t^c}\mathbf{U}^\top - \hat{\mathbf{U}}_{\Omega_t^c}(\hat{\mathbf{U}}_{\Omega_t}^\top \hat{\mathbf{U}}_{\Omega_t})^{-1}\hat{\mathbf{U}}_{\Omega_t}^\top \mathbf{U}_{\Omega_t}\mathbf{U}^\top\|_2 L \qquad \text{(Since } \|\mathbf{U}\alpha\| \leq L\text{)} \\
&= \|\mathbf{U}_{\Omega_t^c}\mathbf{U}^\top - \hat{\mathbf{U}}_{\Omega_t^c}(\hat{\mathbf{U}}_{\Omega_t}^\top \hat{\mathbf{U}}_{\Omega_t})^{-1}\hat{\mathbf{U}}_{\Omega_t}^\top (\mathbf{U}_{\Omega_t} - \hat{\mathbf{U}}_{\Omega_t} + \hat{\mathbf{U}}_{\Omega_t})\mathbf{U}^\top\|_2 L \\
&= \|\mathbf{U}_{\Omega_t^c}\mathbf{U}^\top - \hat{\mathbf{U}}_{\Omega_t^c}\mathbf{U}^\top - \hat{\mathbf{U}}_{\Omega_t^c}(\hat{\mathbf{U}}_{\Omega_t}^\top \hat{\mathbf{U}}_{\Omega_t})^{-1}\hat{\mathbf{U}}_{\Omega_t}^\top (\mathbf{U}_{\Omega_t} - \hat{\mathbf{U}}_{\Omega_t})\mathbf{U}^\top\|_2 L \\
&\leq \|\mathbf{U}_{\Omega_t^c} - \hat{\mathbf{U}}_{\Omega_t^c}\|_2 L + \|\hat{\mathbf{U}}_{\Omega_t^c}(\hat{\mathbf{U}}_{\Omega_t}^\top \hat{\mathbf{U}}_{\Omega_t})^{-1}\hat{\mathbf{U}}_{\Omega_t}^\top (\mathbf{U}_{\Omega_t} - \hat{\mathbf{U}}_{\Omega_t})\mathbf{U}^\top\|_2 L \\
&\qquad\qquad \text{(Using triangle inequality)} \\
&\leq \left(1 + \frac{1}{\lambda_{\min}(\hat{\mathbf{U}}_{\Omega_t}^\top \hat{\mathbf{U}}_{\Omega_t})}\right) L\epsilon_t
\end{aligned}
$$

In order for $(\hat{\mathbf{U}}_{\Omega_t}^\top \hat{\mathbf{U}}_{\Omega_t})^{-1}$ to be invertible, we need to observe at least m entries. Lemma 9 helps define an event as $\mathcal{E}_2(\delta)$ where at least m entries are observed with probability $1-\delta$. Simplifying further:

$$
\begin{aligned}
\|E_t\|_2 &\leq \left(1 + \frac{1}{\lambda_{\min}(\hat{\mathbf{U}}_{\Omega_t}^\top \hat{\mathbf{U}}_{\Omega_t})}\right) L\epsilon_t \\
&\leq \left(1 + \frac{2}{p}\right) L\epsilon_t \qquad \text{(Using Lemma 10)}
\end{aligned}
$$

with p satisfying the following condition:

$$p > \min\left\{\max\left\{\frac{m}{d} + \sqrt{\frac{\log(1/\delta)}{2d}}, 16\left(\frac{\mu^2 m}{d}\right)\log\left(\frac{m}{\delta}\right)\right\}, 1\right\}$$

This can be further simplified to:

$$p > \min\left\{16\left(\frac{\mu^2 m}{d}\right)\log\left(\frac{m}{\delta}\right), 1\right\},$$

as $\mu \in \left[1, \sqrt{\frac{d}{m}}\right]$.

$$\mathbf{X}_t \in \mathcal{C}_{X,t} = \prod_{i=1}^{K} \mathcal{C}_{X_{t,i}}$$

Now, let's consider the probabilistic guarantees:

If each individual constraint set $\mathcal{C}_{X_{t,i}}$ holds with probability at least $1 - \delta/T$, then the probability that the Cartesian product holds can be analyzed as follows:

1. The Cartesian product holds if and only if each $X_{t,i}$ is in its corresponding set $\mathcal{C}_{X_{t,i}}$.
2. The probability that $(X_{t,1}, X_{t,2}, ..., X_{t,K})$ is not in the Cartesian product is equal to the probability that at least one $X_{t,i}$ is not in its corresponding $\mathcal{C}_{X_{t,i}}$.
3. Using the union bound, we can say:

$$\mathbb{P}(\mathbf{X}_t \notin \prod_{i=1}^{K} \mathcal{C}_{X_{t,i}}) \leq \sum_{i=1}^{K} \mathbb{P}(\mathbf{X}_t \notin \mathcal{C}_{X_{t,i}})$$

4. Since each individual set holds with probability at least $1 - \delta/TK$, we have:

$$\mathbb{P}(\mathbf{X}_t \notin \mathcal{C}_{X_{t,i}}) \leq \delta/TK \text{ for each } i$$

5. Using union bound for all i:

$$\mathbb{P}(\mathbf{X}_t \notin \prod_{i=1}^{K} \mathcal{C}_{X_{t,i}}) \leq \cdot\delta/T$$

6. Using union bound for all $t \in [T]$:

$$\mathbb{P}(\mathbf{X}_t \notin \prod_{i=1}^{K} \mathcal{C}_{X_{t,i}}) \leq \cdot\delta$$

7. Therefore, the probability that $\mathbf{X}_t$ is in the Cartesian product is:

$$\mathbb{P}(\mathbf{X}_t \in \prod_{i=1}^{K} \mathcal{C}_{X_{t,i}}) \geq 1 - \delta$$

So, if each individual constraint set $\mathcal{C}_{X_{t,i}}$ holds with probability at least $1 - \delta/TK$, then the Cartesian product of these sets will hold with probability at least $1 - K\delta$, $\forall i \in [K]$ and $\forall t \in [T]$.

4.12.1 Supporting Lemmas

Lemma 8 *If θ_1 is the largest principal angle between the subspaces spanned by $\hat{\mathbf{U}}$ and $\mathbf{U}$, then:*

$$\|\hat{\mathbf{U}} - \mathbf{U}\|_2 \leq 2 \sin \theta_1.$$

Proof:

- The principal angles between the subspaces spanned by $\hat{\mathbf{U}}$ and $\mathbf{U}$ can be found using the singular value decomposition (SVD) of $\hat{\mathbf{U}}^\top \mathbf{U}$. Let the SVD be $\hat{\mathbf{U}}^\top \mathbf{U} = \mathbf{V}\boldsymbol{\Gamma}\mathbf{W}^\top$, where Γ contains the singular values, and $\mathbf{V}$ and $\mathbf{W}$ are orthogonal matrices.
- The singular values in $\boldsymbol{\Gamma}$ are equal to the cosines of the principal angles between the subspaces spanned by $\hat{\mathbf{U}}$ and $\mathbf{U}$. Let σ_1 be the largest singular value, which corresponds to the largest principal angle θ_1. Then, $\sigma_1 = \cos \theta_1$.

- The operator norm of the difference between $\hat{\mathbf{U}}$ and U can be expressed as:

$$\|\hat{\mathbf{U}} - \mathbf{U}\|_2 = \sup_{\|x\|_2=1} \|(\hat{\mathbf{U}} - \mathbf{U})x\|_2.$$

Using the fact that $\hat{\mathbf{U}}^\top \hat{\mathbf{U}} = \mathbf{I}$ and $\mathbf{U}^\top \mathbf{U} = \mathbf{I}$, we can rewrite the operator norm as:

$$\|\hat{\mathbf{U}} - \mathbf{U}\|_2 = \sup_{\|x\|_2=1} \sqrt{x^\top (\hat{\mathbf{U}} - \mathbf{U})^\top (\hat{\mathbf{U}} - \mathbf{U})x}$$

Expanding the expression inside the square root, we get:

$$\|\hat{\mathbf{U}} - \mathbf{U}\|_2 = \sup_{\|x\|_2=1} \sqrt{x^\top (\mathbf{I} - \hat{\mathbf{U}}^\top \mathbf{U} - \mathbf{U}^\top \hat{\mathbf{U}} + \mathbf{U}^\top \mathbf{U})x}$$

Simplifying the expression using the fact that $\hat{\mathbf{U}}^\top \mathbf{U} = \mathbf{V}\mathbf{\Gamma}\mathbf{W}^\top$, we get:

$$\|\hat{\mathbf{U}} - \mathbf{U}\|_2 = \sup_{\|x\|_2=1} \sqrt{x^\top (2\mathbf{I} - 2\mathbf{V}\mathbf{\Gamma}\mathbf{W}^\top)x}.$$

Combining like terms and using the fact that $\mathbf{\Gamma}$ contains the singular values, we get:

$$\|\hat{\mathbf{U}} - \mathbf{U}\|_2 = \sqrt{2 - 2\sigma_1},$$

where, σ_1 is the largest singular value. Using the fact that $\sigma_1 = \cos(\theta_1)$, we can rewrite the expression as:

$$\|\hat{\mathbf{U}} - \mathbf{U}\|_2 = \sqrt{2 - 2\cos\theta_1} \leq \sqrt{2(1 - \cos^2\theta_1)}.$$

Simplifying the expression, we get:

$$\|\hat{\mathbf{U}} - \mathbf{U}\|_2 \leq 2\sin\theta_1.$$

■

Lemma 9 *Let $\Omega_{t,i}$ be the observed index set at time t of the i^{th} action vector $X_{t,i}$. The for all $t \in [T]$ and $i \in [K]$, if $p \geq \frac{m}{d} + \sqrt{\frac{\log(TK/\delta)}{2d}}$, then $\mathbb{P}\{|\Omega_{t,i}| \geq m\} \geq 1 - \delta$.*

Proof: By Hoeffding's Inequality, we have:

$$\mathbb{P}\left\{\sum_{i=1}^{d} s^{(i)} - \mathbb{E}\left[\sum_{i=1}^{d} s^{(i)}\right] \leq -\varepsilon\right\} \leq \exp\left(-\frac{2\varepsilon^2}{d}\right)$$

Since $\mathbb{E}\left[\sum_i s^{(i)}\right] = pd$, we can choose $\varepsilon = dp - m$ to get:

$$\mathbb{P}\left\{\sum_{i=1}^{d} s^{(i)} \leq m\right\} \leq \exp\left(-\frac{2(dp-m)^2}{d}\right)$$

To ensure that the event happens with low probability, we want the right-hand side of the inequality to be close to 0. Let's set it to δ/TK, where δ is a small positive value:

$$\exp\left(-\frac{2(m-dp)^2}{d}\right) \leq \frac{\delta}{TK}$$

Taking the natural logarithm of both sides and solving for p, we get:

$$p \geq \frac{m}{d} + \sqrt{\frac{\log(TK/\delta)}{2d}}$$

Therefore, if we choose p such that $p \geq \frac{m}{d} + \sqrt{\frac{\log(TK/\delta)}{2d}}$, we can say that:

$$\mathbb{P}\left\{\sum_{i=1}^{d} s^{(i)} \geq m\right\} \geq 1 - \frac{\delta}{TK}.$$

By applying the union bound over all time steps $t \in [T]$ and for all indices $i \in [K]$, we get the final result. ■

Lemma 10 *Let $\hat{\mathbf{U}}_t \in \mathbb{R}^{d\times m}$ be an estimate of the true subspace $\mathbf{U}$ such that $\|\hat{\mathbf{U}}_{\Omega_t} - \mathbf{U}\| \leq 2\epsilon_t$, where we set $\epsilon_t \leq p/32$. Suppose we observe an index set $\Omega_t \subset \{1, 2, \ldots, d\}$ and let $\hat{\mathbf{U}}_{\Omega_t}$ be sub-matrix formed by selecting the rows of $\hat{\mathbf{U}}_t$ corresponding to the index set Ω_t. The minimum eigenvalue of the matrix $\hat{\mathbf{U}}_{\Omega_t}^\top \hat{\mathbf{U}}_{\Omega_t}$ can be bounded below by*

$$\lambda_{\min}(\hat{\mathbf{U}}_{\Omega_t}^\top \hat{\mathbf{U}}_{\Omega_t}) \geq p/2$$

with probability at least $1 - \delta$ for all $t \in [t_b, T]$ provided

$$p > 16\frac{\mu^2 m}{d}\log\left(\frac{mTK}{\delta}\right).$$

Proof: We can write $\mathcal{R}_\Omega$ as:

$$\mathcal{R}_{\Omega_t} = \sum_{i=1}^{d} \mathbf{1}(i \in \Omega_t)(\hat{\mathbf{U}}^i)^\top \hat{\mathbf{U}}^i$$

We can bound each of these independent random matrices $(\hat{\mathbf{U}}^i_{\Omega_t})^\top \hat{\mathbf{U}}^i_{\Omega_t}$ by:

$$\begin{aligned}
\|(\hat{\mathbf{U}}^i_{\Omega_t})^\top \hat{\mathbf{U}}^i_{\Omega_t}\|_2 &\leq \|\hat{\mathbf{U}}^i_{\Omega_t}\|_2^2 && \text{(Frobenius norm of } i\text{−th row)} \\
&\leq \|\hat{\mathbf{U}}^i_{\Omega_t} - \mathbf{U}^i_{\Omega_t}\|_2^2 + \|\mathbf{U}^i_{\Omega_t}\|_2^2 \\
&\leq \|\hat{\mathbf{U}} - \mathbf{U}\|_2^2 + \max_i \|\mathbf{U}^i\|_2^2 \\
&\leq 2\epsilon_t + \frac{\mu^2 m}{d} && \text{(Using Lemma 8)}
\end{aligned}$$

The expected value of $\mathcal{R}_{\Omega_t}$ is

$$\mathbb{E}[\mathcal{R}_{\Omega_t}] = p\hat{\mathbf{U}}^\top \mathbf{I}\hat{\mathbf{U}} = p\mathbf{I}_m$$

Therefore, the minimum eigenvalue of $\mathbb{E}[\mathcal{R}_{\Omega_t}]$ is

$$\mu_{\min} = p$$

Applying Matrix Chernoff Inequality we get:

$$\mathbb{P}\left(\lambda_{\min}(\mathcal{R}_{\Omega_t}) \leq \varepsilon p\right) \leq m \exp\left(-\frac{(1-\varepsilon)^2 p}{2\left(\frac{\mu^2 m}{d} + 2\epsilon_t\right)}\right)$$

We can also write this as:

$$\mathbb{P}\left(\lambda_{\min}(\mathcal{R}_{\Omega_t}) \geq \varepsilon p\right) \geq 1 - m \exp\left(-\frac{(1-\varepsilon)^2 p}{2\left(\frac{\mu^2 m}{d} + 2\epsilon_t\right)}\right)$$

We let $\varepsilon = 1/2$, this give us:

$$\mathbb{P}\left(\lambda_{\min}(\mathcal{R}_{\Omega_t}) \geq p/2\right) \geq 1 - m \exp\left(-\frac{p}{8\left(\frac{\mu^2 m}{d} + 2\epsilon_t\right)}\right)$$

Here, ϵ_t decreases with $\frac{1}{\sqrt{t}}$. Additionally, we set $\epsilon_t \leq p/32$.

$$\mathbb{P}\left(\lambda_{\min}(\mathcal{R}_{\Omega_t}) \geq p/2\right) \geq 1 - m\exp\left(-\frac{p}{8\left(\frac{\mu^2 m}{d} + 2\epsilon_t\right)}\right) \geq 1 - m\exp\left(-\frac{p}{8\left(\frac{\mu^2 m}{d} + \frac{p}{16}\right)}\right)$$

We can take $m\exp\left(-\frac{p}{8\left(\frac{\mu^2 m}{d}+\frac{p}{16}\right)}\right) \leq \frac{\delta}{TK}$ to get a bound on p:

$$p \geq 16\frac{\mu^2 m}{d}\log\left(\frac{mTK}{\delta}\right).$$

■

4.13 Confidence Set Construction for Model Parameter

At time index $t \in [T]$, we recall the definition of the following quantities:

- Estimated model parameter at time t: $\hat{\theta}_t = A_t^{\dagger}(\hat{\mathbf{P}}_t\hat{\mathbf{X}}_{t-1}\mathbf{r}_{t-1})$.
- $A_t = \hat{\mathbf{P}}_t\left(\hat{\Sigma}_{t-1} + \lambda I_d\right)\hat{\mathbf{P}}_t$.
- Imputed covariance matrix is given by $\hat{\Sigma}_t = \sum_t \hat{X}_t\hat{X}_t^{\top}$
- Estimated projection matrix is $\hat{\mathbf{P}}_t = \hat{\mathbf{U}}_t\hat{\mathbf{U}}_t^{\top}$

Let $\mathbf{S}_t = \sum_{i=1}^{t}\hat{\mathbf{P}}_t\hat{X}_{i-1}\eta_{i-1} = \hat{\mathbf{P}}_t\hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}$. We get the following:

$$\begin{aligned}
\hat{\theta}_t &= A_t^{\dagger}(\hat{\mathbf{P}}_t\hat{\mathbf{X}}_{t-1}\mathbf{r}_{t-1}) \\
&= A_t^{\dagger}\hat{\mathbf{P}}_t\hat{\mathbf{X}}_{t-1}(\mathbf{X}_{t-1}^{\top}P\theta^* + \boldsymbol{\eta}_{t-1}) \\
&= A_t^{\dagger}\mathbf{S}_t + A_t^{\dagger}\hat{\mathbf{P}}_t\hat{\mathbf{X}}_{t-1}\mathbf{X}_{t-1}^{\top}P\theta^* \\
&= A_t^{\dagger}\mathbf{S}_t + A_t^{\dagger}\hat{\mathbf{P}}_t\hat{\mathbf{X}}_{t-1}(\mathbf{X}_{t-1} - \hat{\mathbf{X}}_{t-1})^{\top}P\theta^* + A_t^{\dagger}\hat{\mathbf{P}}_t\hat{\Sigma}_{t-1}P\theta^* \\
&= A_t^{\dagger}\mathbf{S}_t + A_t^{\dagger}(\hat{\mathbf{P}}_t\hat{\Sigma}_{t-1}(P - \hat{\mathbf{P}}_t + \hat{\mathbf{P}}_t) + \lambda\hat{\mathbf{P}}_t - \lambda\hat{\mathbf{P}}_t)\theta^* + A_t^{\dagger}\hat{\mathbf{P}}_t\hat{\mathbf{X}}_{t-1}(\mathbf{X}_{t-1} - \hat{\mathbf{X}}_{t-1})^{\top}P\theta^* \\
&= A_t^{\dagger}\mathbf{S}_t + A_t^{\dagger}(\hat{\mathbf{P}}_t\hat{\Sigma}_{t-1}(P - \hat{\mathbf{P}}_t))\theta^* + \theta^* - \lambda A_t^{\dagger}\hat{\mathbf{P}}_t\theta^* + A_t^{\dagger}\hat{\mathbf{P}}_t\hat{\mathbf{X}}_{t-1}(\mathbf{X}_{t-1} - \hat{\mathbf{X}}_{t-1})^{\top}P\theta^*.
\end{aligned}$$

Using this, we derive the following for $x = A_t(\theta_t - \theta^*)$:

$$\begin{aligned} x^\top \hat{\theta}_t - x^\top \theta^* &= x^\top A_t^\dagger \mathbf{S}_t + x^\top A_t^\dagger (\hat{\mathbf{P}}_t \hat{\Sigma}_{t-1}(P - \hat{\mathbf{P}}_t))\theta^* - \lambda x^\top A_t^\dagger \hat{\mathbf{P}}_t \theta^* \\ &\quad + x^\top A_t^\dagger \hat{\mathbf{P}}_t \hat{\mathbf{X}}_{t-1}(\mathbf{X}_{t-1} - \hat{\mathbf{X}}_{t-1})^\top P\theta^* \\ &= \langle x^\top, \mathbf{S}_t \rangle_{A_t^\dagger} + \langle x^\top, (\hat{\mathbf{P}}_t \hat{\Sigma}_{t-1}(P - \hat{\mathbf{P}}_t))\theta^* \rangle_{A_t^\dagger} - \lambda \langle x^\top, \hat{\mathbf{P}}_t \theta^* \rangle_{A_t^\dagger} \\ &\quad + \langle x^\top, \hat{\mathbf{P}}_t \hat{\mathbf{X}}_{t-1}(\mathbf{X}_{t-1} - \hat{\mathbf{X}}_{t-1})^\top P\theta^* \rangle_{A_t^\dagger}. \end{aligned}$$

Using Cauchy-Schwarz inequality, we can upper bound the magnitude of the difference as follows:

$$\begin{aligned} |x^\top \hat{\theta}_t - x^\top \theta^*| &\leq \|x\|_{A_t^\dagger} (\|\mathbf{S}_t\|_{A_t^\dagger} + \|\hat{\mathbf{P}}_t \hat{\Sigma}_{t-1}(P - \hat{\mathbf{P}}_t))\theta^*\|_{A_t^\dagger} + \lambda \|\hat{\mathbf{P}}_t \theta^*\|_{A_t^\dagger} \\ &\quad + \|\hat{\mathbf{P}}_t \hat{\mathbf{X}}_{t-1}(\mathbf{X}_{t-1} - \hat{\mathbf{X}}_{t-1})^\top P\theta^*\|_{A_t^\dagger}) \\ &\leq \|x\|_{A_t^\dagger} (\|\mathbf{S}_t\|_{A_t^\dagger} + \|(A_t^\dagger)^{1/2} \hat{\mathbf{P}}_t \hat{\Sigma}_{t-1}\|_2 \|P - \hat{\mathbf{P}}_t\|_2 \|\theta^*\|_2 + \sqrt{\lambda}\|\theta^*\|_2 \\ &\quad + \|(A_t^\dagger)^{1/2} \hat{\mathbf{P}}_t \hat{\mathbf{X}}_{t-1}(\mathbf{X}_{t-1} - \hat{\mathbf{X}}_{t-1})^\top\|_2 \|\theta^*\|_2). \end{aligned}$$

Substituting the value of $x = A_t(\theta_t - \theta^*)$ and using the fact that $\|\theta^*\|_2 \leq S$, we get:

$$\begin{aligned} \|\hat{\theta}_t - \theta^*\|_{A_t} \leq \underbrace{\|\mathbf{S}_t\|_{A_t^\dagger}}_{\text{Term 1}} + \underbrace{S\|(A_t^\dagger)^{1/2} \hat{\mathbf{P}}_t \hat{\Sigma}_{t-1}\|_2 \|P - \hat{\mathbf{P}}_t\|_2}_{\text{Term 2}} \\ + \underbrace{\|(A_t^\dagger)^{1/2} \hat{\mathbf{P}}_t \hat{\mathbf{X}}_{t-1}(\mathbf{X}_{t-1} - \hat{\mathbf{X}}_{t-1})\|_2 \|\theta^*\|_2}_{\text{Term 3}} + \sqrt{\lambda}\|\theta^*\|_2. \end{aligned} \tag{4.27}$$

We will now analyze each of the terms in the above inequality and derive appropriate upper bounds for them.

4.13.1 Term 1: $\|\mathbf{S}_t\|_{A_t^\dagger}$

We state the following Theorem to bound this term:

Theorem 7 *For any $\delta > 0$, with probability at least $1 - \delta$, for all $t \geq 1$,*

$$\|\mathbf{S}_t\|_{A_t^\dagger}^2 \leq 2R^2 \log\left(\frac{\det(B_t)^{1/2} \det(\lambda \mathbf{I}_m)^{-1/2}}{\delta}\right).$$

Here, B_t be a symmetric matrix such that $A_t = \hat{\mathbf{U}}_t B_t \hat{\mathbf{U}}_t^\top$. This gives us $B_t = \hat{\mathbf{U}}_t^\top \hat{\Sigma}_{t-1} \hat{\mathbf{U}}_t + \lambda I_m$.

Proof: In traditional linear bandits, the standard approach involves demonstrating that M_t^ν forms an adaptive supermartingale. This allows the application of a maximal inequality, ensuring the desired bound holds uniformly for all t. However, in our setting, M_t^ν is not a supermartingale (see, Lemma 15), which prevents us from directly leveraging the maximal inequality.

Instead, we establish an upper bound on $\mathbb{E}[M_t^\nu]$ for each t. We then apply Markov's inequality, followed by a union bound across all time steps $t \in [T]$, to derive our result, ensuring it holds uniformly for all t.

1. For an arbitrary $\nu \in \mathbb{R}^d$, define M_t^ν:

$$M_t^\nu := \exp\left(\nu^\top \mathbf{S}_t - \frac{1}{2}\nu^\top \hat{\mathbf{P}}_t \hat{\Sigma}_{t-1} \hat{\mathbf{P}}_t \nu\right)$$

   M_t^ν is non-negative. Define $M_t = \mathbb{E}_\nu[M_t^\nu | \mathcal{F}_\infty]$, where $\mathcal{F}_\infty$ is the tail $\sigma-$algebra of the filtration i.e. the $\sigma-$algebra generated by the union of the all events in the filtration.

$$M_t = \int_{\mathbb{R}^d} M_t^\nu f(\nu) d\nu.$$

   Here, $f(\nu)$ is PDF of ν.

2. Show that $\mathbb{E}[M_t] \leq 1$.We can see that $M_0 \leq 1$

   Lemma 11 $\mathbb{E}[M_t] \leq 1$ *for all* $t \geq 0$.

   Proof:

$$\begin{aligned}\mathbb{E}[M_t] &= \mathbb{E}\left[\int_{\mathbb{R}^d} M_t^\nu f(\nu) d\nu\right] \\ &= \int_{\mathbb{R}^d} \mathbb{E}\left[M_t^\nu\right] f(\nu) d\nu\end{aligned}$$

Focusing on the expectation term inside the integral. We can write using law of total expectation: $\mathbb{E}\left[M_t^\nu\right] = \mathbb{E}\left[\mathbb{E}_{\eta_{t-1}}[M_t^\nu|\mathcal{F}_{t-1}]\right]$. We define the filtration $\mathcal{F}_t := \{X_1, \eta_1, X_2, \eta_2, \dots X_{t-1}, \eta_{t-1}, X_t\}$.

$$
\begin{aligned}
\mathbb{E}_{\eta_{t-1}}[M_t^\nu|\mathcal{F}_{t-1}] &= \mathbb{E}_{\eta_{t-1}}\left[\exp\left(\nu^\top \mathbf{S}_t - \frac{1}{2}\nu^\top \hat{\mathbf{P}}_t \hat{\Sigma}_{t-1}\hat{\mathbf{P}}_t\nu\right)\middle|\mathcal{F}_{t-1}\right] \\
&= \mathbb{E}_{\eta_{t-1}}\left[\exp\left(\nu^\top \mathbf{S}_{t-1} - \frac{1}{2}\nu^\top \hat{\mathbf{P}}_t \hat{\Sigma}_{t-2}\hat{\mathbf{P}}_t\nu\right)\right. \\
&\quad \left.\times \exp\left(\nu^\top \hat{\mathbf{P}}_t \hat{X}_{t-1}\eta_{t-1}\right)\exp\left(-\frac{1}{2}\nu^\top \hat{\mathbf{P}}_t \hat{X}_{t-1}\hat{X}_{t-1}^\top \hat{\mathbf{P}}_t\nu\right)\middle|\mathcal{F}_{t-1}\right] \\
&= \mathbb{E}_{\eta_{t-1}}\left[M_{t-1}^\nu \exp\left(\nu^\top \hat{\mathbf{P}}_t \hat{X}_{t-1}\eta_{t-1} - \frac{1}{2}\nu^\top \hat{\mathbf{P}}_t \hat{X}_{t-1}\hat{X}_{t-1}^\top \hat{\mathbf{P}}_t\nu\right)\middle|\mathcal{F}_{t-1}\right] \\
&= M_{t-1}^\nu \mathbb{E}_{\eta_{t-1}}\left[\exp\left(\nu^\top \hat{\mathbf{P}}_t \hat{X}_{t-1}\eta_{t-1}\right)\middle|\mathcal{F}_{t-1}\right]\exp\left(-\frac{1}{2}\nu^\top \hat{\mathbf{P}}_t \hat{X}_{t-1}\hat{X}_{t-1}^\top \hat{\mathbf{P}}_t\nu\right) \\
&\overset{(a)}{\leq} M_{t-1}^\nu \exp\left(\frac{1}{2}\nu^\top \hat{\mathbf{P}}_t \hat{X}_{t-1}\hat{X}_{t-1}^\top \hat{\mathbf{P}}_t\nu\right)\exp\left(-\frac{1}{2}\nu^\top \hat{\mathbf{P}}_t \hat{X}_{t-1}\hat{X}_{t-1}^\top \hat{\mathbf{P}}_t\nu\right) \\
&\leq M_{t-1}^\nu.
\end{aligned}
$$

Here, inequality (a) is due to the fact that η_{t-1} is conditionally $1-$subgaussian means that,

$$
\mathbb{E}\left[\exp\left(\eta_{t-1}\langle \nu, \hat{\mathbf{P}}_t \hat{X}_{t-1}\rangle\right) \mid \mathcal{F}_{t-1}\right] \leq \exp\left(\langle \nu, \hat{\mathbf{P}}_t \hat{X}_{t-1}\rangle^2\right).
$$

Going recursively we get $\mathbb{E}_{\eta_{t-2}}[M_{t-1}^\nu|\mathcal{F}_{t-2}] \leq M_{t-2}^\nu$. Finally, we get $\mathbb{E}_{\eta_{t-1}}[M_t^\nu|\mathcal{F}_{t-1}] \leq M_0^\nu \leq 1$. Therefore, we can write that $\mathbb{E}[M_t] \leq 1$. ∎

3. Compute $M_t = \int_{\mathbb{R}^d} \exp\left(\nu^\top \mathbf{S}_t - \frac{1}{2}\nu^\top \hat{\mathbf{P}}_t \hat{\Sigma}_{t-1}\hat{\mathbf{P}}_t\nu\right) f(\nu)d\nu$.

$$
\begin{aligned}
M_t &= \int_{\mathbb{R}^d} \exp\left(\nu^\top \mathbf{S}_t - \frac{1}{2}\nu^\top \hat{\mathbf{P}}_t \hat{\Sigma}_{t-1}\hat{\mathbf{P}}_t\nu\right) f(\nu)d\nu \\
&= \int_{\mathbb{R}^m} \exp\left(\tilde{\nu}^\top \hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1} - \frac{1}{2}\tilde{\nu}^\top \hat{\mathbf{U}}_t^\top \hat{\Sigma}_{t-1}\hat{\mathbf{U}}_t\tilde{\nu}\right) f(\tilde{\nu})d\tilde{\nu} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(Change of integral: } \tilde{\nu} = \hat{\mathbf{U}}_t^\top \nu)
\end{aligned}
$$

Let us denote $\bar{B}_t = \hat{\mathbf{U}}_t^\top \hat{\Sigma}_{t-1} \hat{\mathbf{U}}_t$. Then we can see that:

$$\begin{aligned}
\tilde{\nu}^\top \hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1} \boldsymbol{\eta}_{t-1} - \frac{1}{2} \tilde{\nu}^\top \hat{\mathbf{U}}_t^\top \hat{\Sigma}_{t-1} \hat{\mathbf{U}}_t \tilde{\nu} &= \tilde{\nu}^\top \bar{B}_t \bar{B}_t^{-1} \hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1} \boldsymbol{\eta}_{t-1} - \frac{1}{2} \tilde{\nu}^\top \bar{B}_t \tilde{\nu} \\
&\quad + \frac{1}{2} \| \bar{B}_t^{-1} \hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1} \boldsymbol{\eta}_{t-1} \|_2^2 - \frac{1}{2} \| \bar{B}_t^{-1} \hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1} \boldsymbol{\eta}_{t-1} \|_2^2 \\
&= -\frac{1}{2} \| \tilde{\nu} - \bar{B}_t^{-1} \hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1} \boldsymbol{\eta}_{t-1} \|_{\bar{B}_t}^2 \\
&\quad + \frac{1}{2} \| \hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1} \boldsymbol{\eta}_{t-1} \|_{\bar{B}_t^{-1}}^2
\end{aligned}$$

Resuming our analysis by substituting the result from above, we get:

$$\begin{aligned}
M_t &= \int_{\mathbb{R}^m} \exp\left(-\frac{1}{2} \| \tilde{\nu} - \bar{B}_t^{-1} \hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1} \boldsymbol{\eta}_{t-1} \|_{\bar{B}_t}^2 + \frac{1}{2} \| \hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1} \boldsymbol{\eta}_{t-1} \|_{\bar{B}_t^{-1}}^2 \right) f(\tilde{\nu}) d\tilde{\nu} \\
&= \exp\left(\frac{1}{2} \| \hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1} \boldsymbol{\eta}_{t-1} \|_{\bar{B}_t^{-1}}^2 \right) \int_{\mathbb{R}^m} \exp\left(-\frac{1}{2} \| \tilde{\nu} - \bar{B}_t^{-1} \hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1} \boldsymbol{\eta}_{t-1} \|_{\bar{B}_t}^2 \right) f(\tilde{\nu}) d\tilde{\nu}.
\end{aligned}$$

Using the fact that $f(\tilde{\nu}) = \frac{\exp\left(\frac{1}{2} \tilde{\nu}^\top \bar{K} \tilde{\nu}\right)}{\sqrt{(2\pi)^m \det(\bar{K}^{-1})}}$, where $\bar{K} = \hat{\mathbf{U}}_t^\top K \hat{\mathbf{U}}_t$ and K is some

positive definite matrix.

$$
\begin{aligned}
M_t &= \frac{\exp\left(\frac{1}{2}\|\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|^2_{\bar{B}_t^{-1}}\right)}{\sqrt{(2\pi)^m \det(\bar{K}^{-1})}} \int_{\mathbb{R}^m} \exp\left(-\frac{1}{2}\left(\|\tilde{\nu} - \bar{B}_t^{-1}\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|^2_{\bar{B}_t} + \|\tilde{\nu}\|^2_{\bar{K}}\right)\right) d\tilde{\nu} \\
&\overset{(a)}{=} \frac{\exp\left(\frac{1}{2}\|\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|^2_{\bar{B}_t^{-1}}\right)}{\sqrt{(2\pi)^m \det(\bar{K}^{-1})}} \int_{\mathbb{R}^m} \exp\left(-\frac{1}{2}\left(\|\tilde{\nu} - \bar{B}_t^{-1}\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|^2_{\bar{K}+\bar{B}_t}\right.\right. \\
&\qquad\qquad \left.\left. + \|\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|^2_{\bar{B}_t^{-1}} - \|\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|^2_{(\bar{K}+\bar{B}_t)^{-1}}\right)\right) d\tilde{\nu} \\
&= \frac{\exp\left(\frac{1}{2}\|\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|^2_{(\bar{K}+\bar{B}_t)^{-1}}\right)}{\sqrt{(2\pi)^m \det(\bar{K}^{-1})}} \int_{\mathbb{R}^m} \exp\left(-\frac{1}{2}\|\tilde{\nu} - \bar{B}_t^{-1}\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|^2_{\bar{K}+\bar{B}_t}\right) \\
&= \frac{\exp\left(\frac{1}{2}\|\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|^2_{(\bar{K}+\bar{B}_t)^{-1}}\right)}{\sqrt{(2\pi)^m \det(\bar{K}^{-1})}} \\
&\quad \int_{\mathbb{R}^m} \exp\left(-\frac{1}{2}(\tilde{\nu} - \bar{B}_t^{-1}\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1})^\top(\bar{K}+\bar{B}_t)(\tilde{\nu} - \bar{B}_t^{-1}\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1})\right) \\
&= \frac{\exp\left(\frac{1}{2}\|\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|^2_{(\bar{K}+\bar{B}_t)^{-1}}\right)}{\sqrt{(2\pi)^m \det(\bar{K}^{-1})}} \sqrt{(2\pi)^m \det((\bar{K}+\bar{B}_t))^{-1}} \\
&= \exp\left(\frac{1}{2}\|\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|^2_{(\bar{K}+\bar{B}_t)^{-1}}\right) \sqrt{\frac{\det(\bar{K})}{\det(\bar{K}+\bar{B}_t)}} \\
&\overset{(b)}{=} \exp\left(\frac{1}{2}\|\hat{\mathbf{P}}_t \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|^2_{A_t^\dagger}\right) \sqrt{\frac{\det(\lambda \mathbf{I}_m)}{\det(\lambda \mathbf{I}_m + \bar{B}_t)}} \overset{(c)}{=} \exp\left(\frac{1}{2}\|\mathbf{S}_t\|^2_{A_t^\dagger}\right) \sqrt{\frac{\det(\lambda \mathbf{I}_m)}{\det(B_t)}}
\end{aligned}
$$

Here, inequality (a) follows from:

$$\begin{aligned}
&\left\|\tilde{\nu} - \bar{B}_t^{-1}\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\right\|_{\bar{B}_t}^2 + \|\tilde{\nu}\|_{\bar{K}}^2 \\
&\qquad = \left(\tilde{\nu} - \bar{B}_t^{-1}\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\right)^\top \bar{B}_t \left(\tilde{\nu} - \bar{B}_t^{-1}\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\right) \\
&\qquad\qquad + \tilde{\nu}^\top \bar{K}\tilde{\nu} \\
&\qquad = \tilde{\nu}^\top(\bar{K}+\bar{B}_t)\tilde{\nu} - 2\tilde{\nu}^\top \hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1} \\
&\qquad\qquad + \left(\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\right)^\top \bar{B}_t^{-1}\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1} \\
&\qquad = \tilde{\nu}^\top(\bar{K}+\bar{B}_t)\tilde{\nu} - 2\tilde{\nu}^\top \hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1} + \|\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|_{\bar{B}_t^{-1}}^2 \\
&\qquad = \tilde{\nu}^\top(\bar{K}+\bar{B}_t)\tilde{\nu} - 2\tilde{\nu}^\top \hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1} + \|\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|_{\bar{B}_t^{-1}}^2 \\
&\qquad\qquad + \|\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|_{(\bar{K}+\bar{B}_t)^{-1}}^2 - \|\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|_{(\bar{K}+\bar{B}_t)^{-1}}^2 \\
&\qquad = \|\tilde{\nu} - \bar{B}_t^{-1}\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|_{\bar{K}+\bar{B}_t}^2 + \|\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|_{\bar{B}_t^{-1}}^2 \\
&\qquad\qquad - \|\hat{\mathbf{U}}_t^\top \hat{\mathbf{X}}_{t-1}\boldsymbol{\eta}_{t-1}\|_{(\bar{K}+\bar{B}_t)^{-1}}^2
\end{aligned}$$

Equality (b) follows from choosing, $K = \lambda \mathbf{I}_d$, as a consequence $\bar{K} = \lambda \mathbf{I}_m$. Equality (c) follows from the definition:$B_t = \lambda \mathbf{I}_m + \hat{\mathbf{U}}_t^\top \Sigma_{t-1}\hat{\mathbf{U}}_t = \lambda \mathbf{I}_m + \bar{B}_t$.

4. Using Markov Inequality, using the result $\mathbb{E}[M_t] \leq 1$ derived earlier on, we get our result. To see this, consider:

$$\begin{aligned}
\mathbb{P}\left\{\|\mathbf{S}_t\|_{A_t^\dagger}^2 > 2\log\left(\frac{T}{\delta}\sqrt{\frac{\det(B_t)}{\det(\lambda \mathbf{I}_m)}}\right)\right\} &= \mathbb{P}\left\{\frac{\exp\left(\frac{1}{2}\|\mathbf{S}_t\|_{A_t^\dagger}^2\right)\delta}{T\sqrt{\frac{\det(B_t)}{\det(\lambda \mathbf{I}_m)}}} > 1\right\} \\
&\overset{(d)}{\leq} \mathbb{E}\left[\frac{\exp\left(\frac{1}{2}\|\mathbf{S}_t\|_{A_t^\dagger}^2\right)\delta}{T\sqrt{\frac{\det(B_t)}{\det(\lambda \mathbf{I}_m)}}}\right] \\
&= \mathbb{E}[M_t]\frac{\delta}{T} \overset{(e)}{\leq} \frac{\delta}{T}.
\end{aligned}$$

Here, inequality (d) follows from using Markov's inequality and inequality (e) follows from result derived in Lemma 11.

5. Union bound over T. Define a bad event,

$$\mathcal{E}^c_{\mathbf{S}_t,t}(\delta) = \left\{ \|\mathbf{S}_t\|^2_{A_t^\dagger} > 2R^2 \log\left(\frac{T}{\delta} \cdot \frac{\det(B_t)^{1/2}}{\det(\lambda \mathbf{I}_m)^{1/2}}\right)\right\}.$$

We are interested in the probability of $\bigcup_{t=1}^T \mathcal{E}^c_{\mathbf{S}_t,t}(\delta)$. By union bound

$$\mathbb{P}\left\{\bigcup_{t=1}^T \mathcal{E}^c_{\mathbf{S}_t,t}(\delta)\right\} \leq \sum_{t=1}^T \mathbb{P}\left\{\mathcal{E}^c_{\mathbf{S}_t,t}(\delta)\right\} \leq \delta$$

Therefore the intersection of good events is,

$$\mathbb{P}\left\{\bigcap_{t=1}^T \mathcal{E}_{\mathbf{S}_t,t}(\delta)\right\} = 1 - \mathbb{P}\left\{\bigcup_{t=1}^T \mathcal{E}^c_{\mathbf{S}_t,t}(\delta)\right\} \geq 1 - \delta.$$

We can write with probability at least $1-\delta$, the inequality

$$\|\mathbf{S}_t\|^2_{A_t^\dagger} \leq 2R^2 \log\left(\frac{T}{\delta} \cdot \frac{\det(B_t)^{1/2}}{\det(\lambda \mathbf{I}_m)^{1/2}}\right)$$

holds for all $t \in [T]$.

■

Now all that is left is to bound $\det(B_t)$. We use Lemma 16 to provide the following upper bound which now holds with probability at least $1-4\delta$ for all $t \in [t_b, T]$:

$$\|\mathbf{S}_t\|_{A_t^\dagger} \leq R\sqrt{2\log\left(\frac{1}{\delta}\right) + 8m\log\left(1 + \frac{\zeta L^2(1+\log t)}{m\lambda}\left(1+\frac{2}{p}\right)^2 + \frac{tL^2}{m\lambda}\right)}. \quad (4.28)$$

Here, $\zeta := \frac{12m\lambda_1^2 M^2}{p^2\lambda_m^2 K}\log\left(\frac{2dT}{\delta}\right)$.

4.13.2 Term 2: $\|(A_t^\dagger)^{1/2}\hat{\mathbf{P}}_t\hat{\Sigma}_{t-1}\|_2\|\mathbf{P}-\hat{\mathbf{P}}_t\|_2$

This term can be split into two parts: 1) $\|(A_t^\dagger)^{1/2}\hat{\mathbf{P}}_t\hat{\Sigma}_{t-1}\|_2$ and 2) $\|\mathbf{P}-\hat{\mathbf{P}}_t\|_2$. We define the quantity: $B_{t,i} = \hat{\mathbf{U}}_t^\top \hat{\Sigma}_{i-1}\hat{\mathbf{U}}_t + \lambda I_m$. We can then say that $B_{t,t} = B_t$. We now state the following lemma which will bound the first part of Term 2.

Lemma 12 *Let the action vectors $\|X_t\|_2 \leq L$ for all $t \in [t_b, T]$. Then with probability $1 - 3\delta$,*

$$\|(A_t^{\dagger})^{1/2}\hat{\mathbf{P}}_t\hat{\Sigma}_{t-1}\|_2 \leq 2L\left(1 + \left(1 + \frac{2}{p}\right)\sqrt{\frac{\zeta(1+\log t)}{t\lambda}}\right) \times \sqrt{m\log\left(1 + \frac{\zeta L^2(1+\log t)}{m\lambda}\left(1+\frac{2}{p}\right)^2 + \frac{tL^2}{m\lambda}\right)}.$$

Here, $\zeta := \frac{12m\lambda_1^2M^2}{p^2\lambda_m^2K}\log\left(\frac{2dT}{\delta}\right)$

Proof: Recall the definition of $\hat{\Sigma}_{t-1} = \sum_{i=1}^{t-1}\hat{X}_i\hat{X}_i^{\top}$ and $\zeta := \frac{12m\lambda_1^2M^2}{p^2\lambda_m^2K}\log\left(\frac{2dT}{\delta}\right)$.

Using this, we get:

$$\begin{aligned}
\|(A_t^\dagger)^{1/2}\hat{\mathbf{P}}_t\hat{\Sigma}_{t-1}\|_2 &= \sum_{i=1}^{t}\|(A_t^\dagger)^{1/2}\hat{\mathbf{P}}_t\hat{X}_{i-1}\|_2\|\hat{X}_{i-1}\|_2 \\
&\leq \sum_{i=1}^{t}(\|X_{i-1}\|_2 + \|\hat{X}_{i-1}-X_{i-1}\|_2)\|(A_t^\dagger)^{1/2}\hat{\mathbf{P}}_t\hat{X}_{i-1}\|_2 \\
&\qquad\qquad \text{(Using triangle inequality)} \\
&\leq \sum_{i=1}^{t}\left(\|X_i\|_2 + \left(1+\frac{2}{p}\right)L\epsilon_{i-1}\right)\|(A_t^\dagger)^{1/2}\hat{\mathbf{P}}_t\hat{X}_{i-1}\| \qquad \text{(Using Lemma 5)} \\
&\leq \sum_{i=2}^{t}\left(L + \left(1+\frac{2}{p}\right)L\sqrt{\frac{12m\lambda_1^2M^2\log\left(\frac{2dT}{\delta}\right)}{p^2(i-1)K}}\right)\|\hat{\mathbf{P}}_t\hat{X}_{i-1}\|_{A_t^\dagger} \\
&\qquad\qquad \text{(Using definition of } \epsilon_i \text{ from Equation (4.12))} \\
&\overset{(a)}{=} \sum_{i=2}^{t}\left(L + \left(1+\frac{2}{p}\right)L\sqrt{\frac{12m\lambda_1^2M^2\log\left(\frac{2dT}{\delta}\right)}{p^2(i-1)K}}\right)\|\hat{\mathbf{U}}_t^\top\hat{X}_{i-1}\|_{B_t^{-1}} \\
&\overset{(b)}{\leq} \left(\sqrt{\sum_{i=1}^{t}L^2} + \sqrt{\sum_{i=2}^{t}\left(\left(1+\frac{2}{p}\right)L\sqrt{\frac{12m\lambda_1^2M^2\log\left(\frac{2dT}{\delta}\right)}{p^2(i-1)K}}\right)^2}\right) \qquad (4.29) \\
&\qquad \times\sqrt{\sum_{i=1}^{t}\|\hat{\mathbf{U}}_t^\top\hat{X}_{i-1}\|^2_{B_{t,i-1}^{-1}}} \\
&\overset{(c)}{\leq} 2\left(L\sqrt{t} + \left(1+\frac{2}{p}\right)L\sqrt{\frac{12m\lambda_1^2M^2(1+\log(t-1))\log\left(\frac{2dT}{\delta}\right)}{p^2K\lambda}}\right) \\
&\qquad \times\sqrt{m\log\left(1+\frac{\zeta L^2(1+\log t)}{m\lambda}\left(1+\frac{2}{p}\right)^2+\frac{tL^2}{m\lambda}\right)} \\
&\leq 2\left(L\sqrt{t} + \left(1+\frac{2}{p}\right)L\sqrt{\frac{\zeta(1+\log t)}{\lambda}}\right) \qquad (4.30) \\
&\qquad \times\sqrt{m\log\left(1+\frac{\zeta L^2(1+\log t)}{m\lambda}\left(1+\frac{2}{p}\right)^2+\frac{tL^2}{m\lambda}\right)}.
\end{aligned}$$

■

Here, equality (a) is due to $\hat{X}_{i-1}^\top \hat{\mathbf{U}}_t B_t^{-1} \hat{\mathbf{U}}_t^\top \hat{X}_{i-1} = \hat{X}_{t-1}^\top \hat{\mathbf{P}}_t A_t^\dagger \hat{\mathbf{P}}_t \hat{X}_{t-1}$. Inequality (b) is a consequence of how B_t is constructed: $B_{t,i} = B_{t,i-1} + \hat{\mathbf{U}}_t \hat{X}_{i-1} \hat{X}_{i-1}^\top \hat{\mathbf{U}}_t$. The last inequality (c) follows from using Lemma 17.

The second part of Term 2: $\|\mathbf{P} - \hat{\mathbf{P}}_t\|_2$, directly follows from using the Lemma 4. For completeness we restate the result here. With probability at least $1-\delta$, $\forall t > t_b$, the following holds:

$$\|\hat{\mathbf{P}}_t - \mathbf{P}\|_2 \leq \epsilon_t = \sqrt{\frac{\zeta}{t}}. \tag{4.31}$$

Combining the two, we get the final bound on Term 2 as:

$$\|(A_t^\dagger)^{1/2} \hat{\mathbf{P}}_t \hat{\Sigma}_{t-1}\|_2 \|\mathbf{P} - \hat{\mathbf{P}}_t\|_2 \leq$$
$$8L \left(1 + \left(1 + \frac{2}{p}\right) \sqrt{\frac{\zeta(1+\log t)}{t\lambda}}\right) \sqrt{m\zeta \log\left(1 + \frac{\zeta L^2 (1+\log t)}{m\lambda} \left(1+\frac{2}{p}\right)^2 + \frac{tL^2}{m\lambda}\right)}$$

4.13.3 Term 3: $\|\hat{\mathbf{P}}_t \hat{\mathbf{X}}_{t-1} (\mathbf{X}_{t-1} - \hat{\mathbf{X}}_{t-1})^\top \mathbf{P} \theta_*\|_{A_t^\dagger}$

Lemma 13 *Let the action vectors $\|X_t\|_2 \leq L$ for all $t \in [t_b, T]$. Then with probability $1-3\delta$ the following holds*

$$\|\hat{\mathbf{P}}_t \hat{\mathbf{X}}_{t-1} (\mathbf{X}_{t-1} - \hat{\mathbf{X}}_{t-1})^\top \mathbf{P} \theta_*\|_{A_t^\dagger} \leq$$
$$8SL \left(1 + \frac{2}{p}\right) \sqrt{m\zeta(1+\log t) \log\left(1 + \frac{\zeta L^2 (1+\log t)}{m\lambda} \left(1+\frac{2}{p}\right)^2 + \frac{tL^2}{m\lambda}\right)}.$$

Proof: Recall $\zeta := \frac{12m\lambda_1^2 M^2}{p^2\lambda_m^2 K}\log\left(\frac{2dT}{\delta}\right)$. Using this we get:

$$
\begin{aligned}
&\|\hat{\mathbf{P}}_t\hat{\mathbf{X}}_{t-1}(\mathbf{X}_{t-1}-\hat{\mathbf{X}}_{t-1})^\top\mathbf{P}\theta_*\|_{A_t^\dagger}\\
&\le \|\sum_{i=1}^{t}(A_t^\dagger)^{1/2}\hat{\mathbf{P}}_t\hat{X}_{i-1}(X_{i-1}-\hat{X}_{i-1})^\top\mathbf{P}\theta_*\|_2\\
&\le \sum_{i=1}^{t}\|(A_t^\dagger)^{1/2}\hat{\mathbf{P}}_t\hat{X}_{i-1}(X_{i-1}-\hat{X}_{i-1})^\top\mathbf{P}\theta_*\|_2\\
&\overset{(a)}{\le} \sum_{i=1}^{t}\|(A_t^\dagger)^{1/2}\hat{\mathbf{P}}_t\hat{X}_{i-1}\|_2\|X_{i-1}-\hat{X}_{i-1}\|_2\|\mathbf{P}\theta_*\|_2\\
&\le S\sum_{i=1}^{t}\|(A_t^\dagger)^{1/2}\hat{\mathbf{P}}_t\hat{X}_{i-1}\|_2\|X_{i-1}-\hat{X}_{i-1}\|_2\\
&\overset{(b)}{\le} SL\left(1+\frac{2}{p}\right)\sum_{i=1}^{t}\epsilon_{i-1}\|(A_t^\dagger)^{1/2}\hat{\mathbf{P}}_t\hat{X}_{i-1}\|_2\\
&\overset{(c)}{=} SL\left(1+\frac{2}{p}\right)\sum_{i=2}^{t}\left(\sqrt{\frac{12m\lambda_1^2M^2\log\left(\frac{2dT}{\delta}\right)}{\lambda_m^2p^2(i-1)K}}\right)\|\hat{\mathbf{P}}_t\hat{X}_{i-1}\|_{A_t^\dagger}\\
&\overset{(d)}{=} SL\left(1+\frac{2}{p}\right)\sum_{i=2}^{t}\left(\sqrt{\frac{12m\lambda_1^2M^2\log\left(\frac{2dT}{\delta}\right)}{\lambda_m^2p^2(i-1)K}}\right)\|\hat{\mathbf{U}}_t^\top\hat{X}_{i-1}\|_{B_t^{-1}}\\
&\overset{(e)}{\le} SL\left(1+\frac{2}{p}\right)\sqrt{\sum_{i=2}^{t}\left(\sqrt{\frac{12m\lambda_1^2M^2\log\left(\frac{2dT}{\delta}\right)}{\lambda_m^2p^2(i-1)K}}\right)^2}\sqrt{\sum_{i=1}^{t}\|\hat{\mathbf{U}}_t^\top\hat{X}_{i-1}\|_{B_t^{-1}}^2}\\
&= SL\left(1+\frac{2}{p}\right)\sqrt{\frac{12m\lambda_1^2M^2\log\left(\frac{2dT}{\delta}\right)}{\lambda_m^2p^2K}}\sqrt{\sum_{i=2}^{t}\frac{1}{i-1}}\sqrt{\sum_{i=1}^{t}\|\hat{\mathbf{U}}_t^\top\hat{X}_{i-1}\|_{B_t^{-1}}^2}\\
&\overset{(f)}{=} SL\left(1+\frac{2}{p}\right)\sqrt{\zeta}\sqrt{\sum_{i=2}^{t}\frac{1}{i-1}}\sqrt{\sum_{i=1}^{t}\|\hat{\mathbf{U}}_t^\top\hat{X}_{i-1}\|_{B_t^{-1}}^2}\\
&\le SL\left(1+\frac{2}{p}\right)\sqrt{\zeta}\sqrt{\sum_{i=2}^{t}\frac{1}{i-1}}\sqrt{\sum_{i=1}^{t}\|\hat{\mathbf{U}}_t^\top\hat{X}_{i-1}\|_{B_{t,i-1}^{-1}}^2}\\
&\overset{(g)}{\le} 8SL\left(1+\frac{2}{p}\right)\sqrt{m\zeta(1+\log t)\log\left(1+\frac{\zeta L^2(1+\log t)}{m\lambda}\left(1+\frac{2}{p}\right)^2+\frac{tL^2}{m\lambda}\right)}.
\end{aligned}
$$

Here, equality (a) is using Cauchy-Shwarz inequality. Inequality (b) is a consequence of

using Lemma 5. We get equality (c) by substituting the definition of ϵ_t from Equation 4.12. We get equality (d) by seeing that $\hat{X}_{i-1}^\top \hat{\mathbf{U}}_t B_t^{-1} \hat{\mathbf{U}}_t^\top \hat{X}_{i-1} = \hat{X}_{t-1}^\top \hat{\mathbf{P}}_t A_t^\dagger \hat{\mathbf{P}}_t \hat{X}_{t-1}$. Inequality (e) is using Cauchy-Schwarz again. Equaltiy (f) follows from using the definition of ζ. Last inequality (g) is by using the result in Lemma 17. ■

4.13.4 Confidence Ellipsoid

Putting the bounds on all the terms (Term 1: Equation (4.28), Term 2: 12, Term 3: 13) together in Equation (4.27), we get the ellipsoid $\mathcal{C}_{m,t}$ centered around $\hat{\theta}_t$ which include the true model parameter θ_* with probability at least $1 - 4\delta$ for all $t \in [t_b, T]$:

$$\mathcal{C}_{m,t} = \left\{ \theta \in \mathbb{R}^d : \|\hat{\theta}_t - \theta\|_{A_t} \leq \beta_{t,\delta} \right\}, \text{ where}$$

$$\begin{aligned}
\beta_{t,\delta} := R\sqrt{2\log\left(\frac{1}{\delta}\right) + 8m\log\left(1 + \frac{\zeta L^2(1+\log t)}{m\lambda}\left(1+\frac{2}{p}\right)^2 + \frac{tL^2}{m\lambda}\right)} \\
+ 8L\left(1 + \left(1+\frac{2}{p}\right)\sqrt{\frac{\zeta(1+\log t)}{t\lambda}}\right)\sqrt{m\zeta\log\left(1 + \frac{\zeta L^2(1+\log t)}{m\lambda}\left(1+\frac{2}{p}\right)^2 + \frac{tL^2}{m\lambda}\right)} \\
+ 8SL\left(1+\frac{2}{p}\right)\sqrt{m\zeta(1+\log t)\log\left(1 + \frac{\zeta L^2(1+\log t)}{m\lambda}\left(1+\frac{2}{p}\right)^2 + \frac{tL^2}{m\lambda}\right)} \\
+ S\sqrt{\lambda}.
\end{aligned}$$

Here, $\zeta := \frac{12m\lambda_1^2 M^2}{p^2\lambda_m^2 K}\log\left(\frac{2dT}{\delta}\right)$.

Remark: The dominant term in confidence ellipsoid scales like:

$$\beta_{t,\delta} \in \tilde{\mathcal{O}}\left(\frac{\sqrt{m\zeta}}{p}\right),$$

suppressing logarithm of t terms.

4.13.5 Supporting Lemmas

Lemma 14

1. *Given the sum of square roots of reciprocals, we have the following inequality:*

$$2\sqrt{T+1} \leq \sum_{t=1}^{T} \sqrt{\frac{1}{t}} \leq 2\sqrt{T} - 1$$

2. *Given the sum of reciprocals, we have the following inequality:*

$$\log(T+1) \leq \sum_{t=1}^{T} \frac{1}{t} \leq 1 + \log(T)$$

Lemma 15 *Let* $\mathbf{S}_t = \sum_{i=1}^{t} \hat{\mathbf{P}}_t \hat{X}_{i-1} \eta_{i-1} = \hat{\mathbf{P}}_t \hat{\mathbf{X}}_{t-1} \boldsymbol{\eta}_{t-1}$ *be defined on the filtration* $\mathcal{F}_t = \{X_1, \eta_1, \cdots, X_{t-1}, \eta_{t-1}, X_t\}$ *and* X_t *represents the choice of action vector at time* t*. Then the sequence* $\{\mathbf{S}_t\}_{t=1}^{T}$ *is not a martingale with respect to* $\mathcal{F}_t$*, i.e.,*

$$\mathbb{E}[\mathbf{S}_t | \mathcal{F}_{t-1}] = \mathbf{S}_{t-1}.$$

Proof: To prove that $\mathbf{S}_t$ is a martingale with respect to the filtration $\mathcal{F}_t$, we need to show that:

$$\mathbb{E}[\mathbf{S}_t | \mathcal{F}_{t-1}] = \mathbf{S}_{t-1}.$$

From the definition of $\mathbf{S}_t$, we have:

$$\mathbf{S}_t = \sum_{i=1}^{t} \hat{\mathbf{P}}_t \hat{X}_{i-1} \eta_{i-1},$$

or equivalently:

$$\mathbf{S}_t = \mathbf{S}_{t-1} + \hat{\mathbf{P}}_t \hat{X}_{t-1} \eta_{t-1} + \sum_{i=1}^{t-1} (\hat{\mathbf{P}}_t - \hat{\mathbf{P}}_{t-1}) \hat{X}_{i-1} \eta_{i-1}.$$

We now compute the conditional expectation $\mathbb{E}[\mathbf{S}_t | \mathcal{F}_{t-1}]$. By the linearity of expectation, we have:

$$\mathbb{E}\left[\mathbf{S}_t | \mathcal{F}_{t-1}\right] = \mathbb{E}\left[\mathbf{S}_{t-1} + \hat{\mathbf{P}}_t \hat{X}_{t-1} \eta_{t-1} + \sum_{i=1}^{t-1} (\hat{\mathbf{P}}_t - \hat{\mathbf{P}}_{t-1}) \hat{X}_{i-1} \eta_{i-1} | \mathcal{F}_{t-1}\right].$$

Since $\mathbf{S}_{t-1}$ is $\mathcal{F}_{t-1}$-measurable (as it only depends on information up to time $t-1$), the expectation simplifies to:

$$\mathbb{E}[\mathbf{S}_t|\mathcal{F}_{t-1}] = \mathbf{S}_{t-1} + \mathbb{E}[\hat{\mathbf{P}}_t \hat{X}_{t-1}\eta_{t-1}|\mathcal{F}_{t-1}] + \mathbb{E}\left[\sum_{i=1}^{t-1}(\hat{\mathbf{P}}_t - \hat{\mathbf{P}}_{t-1})\hat{X}_{i-1}\eta_{i-1}|\mathcal{F}_{t-1}\right].$$

We assume that η_{t-1} has zero mean, i.e., $\mathbb{E}[\eta_{t-1}|\mathcal{F}_{t-1}] = 0$. Therefore, the conditional expectation becomes:

$$\begin{aligned}\mathbb{E}[\mathbf{S}_t|\mathcal{F}_{t-1}] &= \mathbf{S}_{t-1} + \mathbb{E}\left[\sum_{i=1}^{t-1}(\hat{\mathbf{P}}_t - \hat{\mathbf{P}}_{t-1})\hat{X}_{i-1}\eta_{i-1}|\mathcal{F}_{t-1}\right] \\ &= \mathbf{S}_{t-1} + \sum_{i=1}^{t-1}(\hat{\mathbf{P}}_t - \hat{\mathbf{P}}_{t-1})\hat{X}_{i-1}\eta_{i-1}\end{aligned}$$

■

Lemma 16 *Let the true action vectors $\|X_t\|_2 \leq L$ for all $t \geq 1$, then with probability at least $1-3\delta$, $\det(B_t) \leq \left(\lambda + \frac{\zeta L^2(1+\log t)}{m}\left(1+\frac{2}{p}\right)^2 + \frac{tL^2}{m}\right)^m$. Here, $\zeta := \frac{12m\lambda_1^2 M^2}{p^2\lambda_m^2 K}\log\left(\frac{2dT}{\delta}\right)$.*

Proof: $\det(B_t) = \det(\hat{\mathbf{U}}_t^\top \hat{\Sigma}_{t-1}\hat{\mathbf{U}}_t + \lambda I_m) = \alpha_1\alpha_2\ldots\alpha_m$, where α_i's are the eigenvalues

of B_t. Using the AM-GM inequality, $\sqrt[m]{\alpha_1\alpha_2\ldots\alpha_m} \leq \frac{1}{m}\sum_{i=1}^{m}\alpha_i$, we get:

$$\begin{aligned}
\alpha_1\alpha_2\ldots\alpha_m &\leq \left(\frac{1}{m}\sum_{i=1}^{m}\alpha_i\right)^m \\
&= \left(\lambda + \frac{1}{m}\sum_{i=1}^{t}\text{tr}\left(\hat{\mathbf{U}}_t^\top \hat{X}_{i-1}\hat{X}_{i-1}^\top \hat{\mathbf{U}}_t\right)\right)^m \\
&\leq \left(\lambda + \frac{1}{m}\sum_{i=1}^{t}\|\hat{X}_{i-1}\|_2^2\right)^m && \text{(Using cyclic property of trace)} \\
&\leq \left(\lambda + \frac{1}{m}\sum_{i=1}^{t}(\|\hat{X}_{i-1} - X_{i-1}\|_2^2 + L^2)\right)^m \\
&\leq \left(\lambda + \frac{1}{m}\sum_{i=1}^{t}\left(\left(1+\frac{2}{p}\right)^2 L^2\epsilon_{i-1}^2 + L^2\right)\right)^m && \text{(Using Lemma 5)} \\
&= \left(\lambda + \frac{1}{m}\sum_{i=2}^{t}\left(12\frac{\lambda_1^2 mM^2L^2}{\lambda_m^2 p^2(i-1)K}\left(1+\frac{2}{p}\right)^2\log\left(\frac{2dT}{\delta}\right) + L^2\right)\right)^m \\
& && \text{(Using Equation (4.12))} \\
&\leq \left(\lambda + \frac{12(\lambda_1 ML)^2(1+\log(t-1))}{\lambda_m^2 p^2 K}\left(1+\frac{2}{p}\right)^2\log\left(\frac{2dT}{\delta}\right) + \frac{tL^2}{m}\right)^m \\
& && \text{(Using Lemma 14)}
\end{aligned}$$

Using the definition of $\zeta := \frac{12m\lambda_1^2M^2}{p^2\lambda_m^2K}\log\left(\frac{2dT}{\delta}\right)$, we can simplify the expression further:

$$\begin{aligned}
\det(B_t) &= \alpha_1\alpha_2\ldots\alpha_m \\
&\leq \left(\lambda + \frac{\zeta L^2(1+\log t)}{m}\left(1+\frac{2}{p}\right)^2 + \frac{tL^2}{m}\right)^m
\end{aligned}$$

■

Lemma 17 *Let the true action vectors, $\|X_t\|_2 \leq L$, $\forall t \in [T]$, then with probability $1-3\delta$, we can say that*

$$\sum_{i=1}^{t}\|\hat{\mathbf{U}}_t^\top \hat{X}_{i-1}\|_{B_{t,i-1}^{-1}}^2 \leq 4m\log\left(1 + \frac{\zeta L^2(1+\log t)}{m\lambda}\left(1+\frac{2}{p}\right)^2 + \frac{tL^2}{m\lambda}\right). \tag{4.32}$$

Proof: We define the quantity: $B_{t,i} = \hat{\mathbf{U}}_t^\top \hat{\Sigma}_{i-1} \hat{\mathbf{U}}_t + \lambda I_m$. We can then say that $B_{t,t} = B_t$. Taking a look at $\det(B_{t,t})$, we see that:

$$\begin{aligned}
\det(B_{t,t}) &= \det(B_{t,t-1}^{1/2}(\mathbf{I}_m + B_{t,t-1}^{-1/2}\hat{\mathbf{U}}_t^\top \hat{X}_{t-1,}\hat{X}_{t-1}^\top \hat{\mathbf{U}}_{t-1} B_{t,t-1}^{-1/2}) B_{t,t-1}^{1/2}) \\
&= \det(B_{t,t-1})(1 + \|\hat{\mathbf{U}}_t^\top \hat{X}_{t-1}\|^2_{B_{t,t-1}^{-1}}) \\
&= \det(\lambda \mathbf{I}_m) \prod_{i=1}^{t} (1 + \|\hat{\mathbf{U}}_t^\top \hat{X}_{i-1}\|^2_{B_{t,i-1}^{-1}}) \\
&= \lambda^m \prod_{i=1}^{t} (1 + \|\hat{\mathbf{U}}_t^\top \hat{X}_{i-1}\|^2_{B_{t,i-1}^{-1}}).
\end{aligned}$$

Therefore, $\sum_{i=1}^{t} \log(1 + \|\hat{\mathbf{U}}_t^\top \hat{X}_{i-1}\|^2_{B_{t,i-1}^{-1}}) = \log\left(\frac{\det(B_t)}{\lambda^m}\right)$. This can upper bounded using Lemma 16. Looking at the term $\|\hat{\mathbf{U}}_t^\top \hat{X}_{i-1}\|^2_{B_{t,i-1}^{-1}}$ we can bound this by

$$\begin{aligned}
\|\hat{\mathbf{U}}_t^\top \hat{X}_{i-1}\|^2_{B_{t,i-1}^{-1}} &\leq \|B_{t,i-1}^{-1}\|_2 \|\hat{X}_{i-1}\|^2 \\
&\leq \frac{1}{\lambda}\left(\|\hat{X}_{i-1} - X_{i-1}\|_2^2 + \|X_{i-1}\|_2^2\right) \\
&\leq \frac{1}{\lambda}\left(\left(1 + \frac{2}{p}\right)^2 L^2 \epsilon_{i-1}^2 + \|X_{i-1}\|_2^2\right) \\
&\leq \frac{1}{\lambda}\left(\frac{12m(\lambda_1 ML)^2}{\lambda_m^2 p^2 (i-1)K}\left(1 + \frac{2}{p}\right)^2 \log\left(\frac{2dT}{\delta}\right) + L^2\right) \\
&\overset{(1)}{\leq} \frac{1}{\lambda}\left(\frac{\zeta L^2}{(i-1)}\left(1 + \frac{2}{p}\right)^2 + L^2\right)
\end{aligned}$$

The inequality (1) holds with probability $1 - 3\delta$. Now, as the quantity $\|\hat{\mathbf{U}}_t^\top \hat{X}_{i-1}\|^2_{B_{t,i-1}^{-1}}$ is positive and upper bounded, if $\lambda \geq \max\{1, \zeta L^2\left(1 + \frac{2}{p}\right)^2 + L^2\}$, we can write: $\|\hat{\mathbf{U}}_t^\top \hat{X}_{i-1}\|^2_{B_{t,i-1}^{-1}} \leq 4\log(1 + \|\hat{\mathbf{U}}_t^\top \hat{X}_{i-1}\|^2_{B_{t,i-1}^{-1}})$. This follows from

$x \leq 4\log(1+x)$, which holds when $x \in [0,1]$.

$$\begin{aligned}\sum_{i=1}^{t} \|\hat{\mathbf{U}}_t^\top \hat{X}_{i-1}\|^2_{B_{t,i-1}^{-1}} &\leq 4\sum_{i=1}^{t} \log\left(1 + \|\hat{\mathbf{U}}_t^\top \hat{X}_{i-1}\|^2_{B_{t,i-1}^{-1}}\right) \\ &= 4\log\left(\frac{\det(B_t)}{\lambda^m}\right) \\ &\overset{(a)}{\leq} 4m\log\left(1 + \frac{\zeta L^2(1+\log t)}{m\lambda}\left(1+\frac{2}{p}\right)^2 + \frac{tL^2}{m\lambda}\right).\end{aligned}$$

(Using Lemma 16)

■

4.14 Regret Analysis

We recall and state the following events that will help us define a good event on which we calculate regret:

- **Event $\mathcal{E}_1(\delta, t)$: Subspace Distance Bound**

  $$\mathcal{E}_1(\delta, t) = \left\{\text{dist}(\hat{\mathbf{U}}_t, \mathbf{U}) \leq \epsilon_t\right\}$$

  This event holds with probability at least $1-\delta$ and for all $t \geq 1$.

- **Event $\mathcal{E}_2(\delta, t)$: Confidence Set Construction for Action Vectors**

  $$\mathcal{E}_2(\delta, t) = \left\{\mathbf{X}_t \in \prod_{i=1}^{K} \mathcal{C}_{X_{t,i}} \text{ for all } t \in [t_b, T]\right\},$$

  where $\mathcal{C}_{X_{t,i}} = \left\{X \in \mathbb{R}^d : \|\hat{X}_{t,i} - X\|_2 \leq \left(1+\frac{2}{p}\right) L\epsilon_t\right\}$. This event holds with probability at least $1-3\delta$ given that $\mathcal{E}_1(\delta, t)$ occurs.

- **Event $\mathcal{E}_3(\delta, t)$: Confidence Set Construction for Model Parameters**

  $$\mathcal{E}_3(\delta, t) = \left\{\mathcal{C}_{m,t} = \left\{\theta \in \mathbb{R}^d : \|\hat{\theta}_t - \theta\|_{A_t} \leq \beta_{t,\delta}\right\} \text{ for all } t \in [t_b, T]\right\}.$$

  This event holds with probability at least $1-4\delta$ given that $\mathcal{E}_2(\delta, t)$ occurs.

We define the "good" event as $\mathcal{E}_3(\delta, t)$. Given that the good event $\mathcal{E}_3(\delta, t)$ occurs, for all $t \in [t_b, T]$ the instantaneous regret regret l_t can be written as:

$$
\begin{aligned}
l_t &= X_t^{*\top}\theta_* - X_t^\top\theta_* \\
&= (\mathbf{P}X_t^*)^\top\theta_* - (\mathbf{P}X_t)^\top\theta_* \\
&\leq (\tilde{P}_t\tilde{X}_t)^\top\tilde{\theta}_t - (\mathbf{P}X_t)^\top\theta_* \\
&\qquad\qquad \text{(Optimistically choosing } (\tilde{P}_t, \tilde{X}_t, \tilde{\theta}_t) \in \mathcal{C}_{\mathbf{P},t} \times \mathcal{C}_{X,t} \times \mathcal{C}_{m,t}) \\
&= (\tilde{P}_t(\tilde{X}_t - X_t + X_t)^\top\tilde{\theta}_t - (\mathbf{P}X_t)^\top\theta_* \\
&= (\tilde{P}_t(\tilde{X}_t - X_t))^\top\tilde{\theta}_t + (\tilde{P}_tX_t)^\top\tilde{\theta}_t - (\mathbf{P}X_t)^\top\theta_* \\
&= (\tilde{P}_t(\tilde{X}_t - X_t + \hat{X}_t - \hat{X}_t))^\top\tilde{\theta}_t + X_t^\top(\tilde{P}_t - \hat{\mathbf{P}}_t + \hat{\mathbf{P}}_t)\tilde{\theta}_t - X_t^\top(\hat{\mathbf{P}}_t + \mathbf{P} - \hat{\mathbf{P}}_t)\theta_* \\
&= (\tilde{P}_t(\tilde{X}_t - \hat{X}_t))^\top\tilde{\theta}_t + (\tilde{P}_t(\hat{X}_t - X_t))^\top\tilde{\theta}_t + (\hat{\mathbf{P}}_tX_t)^\top(\tilde{\theta}_t - \hat{\theta}_t) + (\hat{\mathbf{P}}_tX_t)^\top(\hat{\theta}_t - \theta_*) \\
&+ ((\hat{\mathbf{P}}_t - \mathbf{P})X_t^\top)\theta_* + ((\tilde{P}_t - \hat{\mathbf{P}}_t)X_t^\top)\tilde{\theta}_t \\
&\leq 2S\|\hat{X}_t - X_t\|_2 + 2\beta_{t,\delta}\|\hat{\mathbf{P}}_tX_t\|_{A_t^\dagger} + 2LS\|\hat{\mathbf{P}}_t - \mathbf{P}\|_2. \\
&\qquad\qquad \text{(Using Lemma 18 and Theorem 4)}
\end{aligned}
$$

Regret accumulated over time T is:

$$
\begin{aligned}
R_T &= t_b + \sum_{t=t_b}^{T} l_t \\
&\leq t_b + \underbrace{2S\sum_{t=t_b}^{T}\|\hat{X}_t - X_t\|_2}_{\text{Regret Term 1}} + \underbrace{2LS\sum_{t=t_b}^{T}\|\hat{\mathbf{P}}_t - \mathbf{P}\|_2}_{\text{Regret Term 2}} + \underbrace{2\sum_{t=t_b}^{T}\beta_{t,\delta}\|\hat{\mathbf{P}}_tX_t\|_{A_t^\dagger}}_{\text{Regret Term 3}}
\end{aligned}
$$

Using Lemmas 19, 20 and 21 to bound each of the terms in above regret expression we obtain the following upper bound which holds with probability at least $1 - 4\delta$:

$$
\begin{aligned}
R_T \leq t_b + CLS\left(1 + \frac{2}{p}\right)\sqrt{\zeta T} + \leq CLS\sqrt{\zeta T} \\
+ C\beta_{T,\delta}\left(L\sqrt{\zeta T} + \sqrt{(1 + \zeta + 2\log(T+1) + 4m\log\mathcal{L})T}\right). \qquad (4.33)
\end{aligned}
$$

where, C is a large positive scalar and we define:

$$\zeta := \frac{12m\lambda_1^2 M^2}{p^2\lambda_m^2 K}\log\left(\frac{2dT}{\delta}\right) \tag{4.34}$$

For large T, the dominant terms in the regret expression is:

$$R_T \in \tilde{\mathcal{O}}\left(\beta_{T,\delta}\sqrt{\zeta T}\right).$$

We know how the confidence ellipsoid $\beta_{T,\delta}$ scales like $\tilde{\mathcal{O}}\left(\frac{\sqrt{m\zeta}}{p}\right)$. Therefore, using this the final regret is given by:

$$\begin{aligned} R_T &\in \tilde{\mathcal{O}}\left(\frac{\zeta\sqrt{mT}}{p}\right) \\ &= \tilde{\mathcal{O}}\left(\frac{m^{3/2}\lambda_1^2\sqrt{T}}{p^3\lambda_m^2 K}\right). \end{aligned}$$

Here, the last equality follows from substituting the expression for ζ from Equation (4.34).

4.14.1 Supporting Lemmas

Lemma 18 *At any round t, let X be a true action vector. If $\nu \in \mathcal{C}_{m,t}$, then*

$$|(\hat{\mathbf{P}}_t X)^\top(\nu - \hat{\theta}_t)| \leq \beta_{t,\delta}\|\hat{\mathbf{P}}_t X\|_{A_t^\dagger}.$$

Proof:

$$\begin{aligned} |(\hat{\mathbf{P}}_t X)^\top(\nu - \hat{\theta}_t)| &= |(\hat{\mathbf{P}}_t X)^\top (A_t^\dagger)^{1/2} A_t^{1/2}(\nu - \hat{\theta}_t)| && ((A_t^\dagger)^{1/2}A_t^{1/2} = \hat{\mathbf{P}}_t) \\ &= |((A_t^\dagger)^{1/2}\hat{\mathbf{P}}_t X)^\top A_t^{1/2}(\nu - \hat{\theta}_t)| \\ &\leq \|A_t^{1/2}(\nu - \hat{\theta}_t)\|_2 \|(A_t^\dagger)^{1/2}\hat{\mathbf{P}}_t X\|_2 \\ &\leq \|\nu - \hat{\theta}_t\|_{A_t}\|\hat{\mathbf{P}}_t X\|_{A_t^\dagger} \\ &\leq \beta_{t,\delta}\|\hat{\mathbf{P}}_t X\|_{A_t^\dagger}. \end{aligned}$$

■

Lemma 19 *Suppose the true action vectors satisfy $\|X_t\|_2 \leq L$ for all $t \geq 1$. Let $\hat{X}_t$ be the imputed action vectors. Then the following holds with probability at least $1 - 3\delta$:*

Proof:

$$\begin{aligned}
\sum_{t=b}^{T} \|\hat{X}_t - X_t\|_2 &= \sum_{t=t_b}^{T} \|E_t\|_2 \\
&\leq L\left(1 + \frac{2}{p}\right) \sum_{t=t_b}^{T} \epsilon_t && \text{(From Lemma 5)} \\
&= L\left(1 + \frac{2}{p}\right) \sum_{t=t_b}^{T} \sqrt{\frac{12m\lambda_1^2 M^2}{p^2 \lambda_m^2 K t} \log\left(\frac{2dT}{\delta}\right)} && \text{(Using Equation (4.12))} \\
&\leq L\left(1 + \frac{2}{p}\right) \sqrt{\zeta}(2\sqrt{T} - 2\sqrt{t_b + 1} + 1) && \text{(Using Equation 4.34)} \\
&\leq CL\left(1 + \frac{2}{p}\right) \sqrt{\zeta T}. && (4.35)
\end{aligned}$$

■

Lemma 20 *Suppose the true action vectors satisfy $\|X_t\|_2 \leq L$ for all $t \geq 1$. Then the following holds with probability at least $1 - \delta$:*

Proof:

$$\begin{aligned}
2L \sum_{t=t_b}^{T} \|\hat{\mathbf{P}}_t - \mathbf{P}\|_2 &\leq 2LS \sum_{t=t_b}^{T} \epsilon_t && \text{(Using Lemma 4)} \\
&\leq 8L \sum_{t=t_b}^{T} \sqrt{\frac{12m\lambda_1^2 M^2}{p^2 \lambda_m^2 K t} \log\left(\frac{2dT}{\delta}\right)} \\
&\leq 8L\sqrt{\zeta}(2\sqrt{T} - 2\sqrt{t_b + 1} + 1) \\
&\leq CL\sqrt{\zeta T} && (4.36)
\end{aligned}$$

■

Lemma 21 (Projected Potential Lemma) *Suppose the true action vectors satisfy* $\|X_t\|_2 \leq L$ *for all* $t \geq 1$. *Let* $A_t = \hat{\mathbf{P}}_t(\hat{\Sigma}_{t-1} + \lambda\mathbf{I})\hat{\mathbf{P}}_t$ *and* $B_t = \hat{\mathbf{U}}_t^\top(\hat{\Sigma}_{t-1} + \lambda\mathbf{I})\hat{\mathbf{U}}_t$, *such that* $\hat{\mathbf{U}}_t B_t \hat{\mathbf{U}}_t^\top = A_t$. *Then, for* $t \in [t_b, T]$, *the following bound holds with probability at least* $1 - 4\delta$*:*

$$2\sum_{t=t_b}^{T} \beta_{t,\delta} \|\hat{\mathbf{P}}_t X_t\|_{A_t^\dagger} \leq C\beta_{T,\delta}\left(L\sqrt{\zeta T} + \sqrt{(1 + \zeta + 2\log(T+1) + 4m\log\mathcal{L})T}\right), \tag{4.37}$$

where, $\mathcal{L} = \log\left(1 + \frac{12\zeta L^2(1+\log t)}{m\lambda}\left(1 + \frac{2}{p}\right)^2 + \frac{tL^2}{m\lambda}\right)$, $\zeta := \frac{12m\lambda_1^2 M^2}{p^2\lambda_m^2 K}\log\left(\frac{2dT}{\delta}\right)$ *and* C *is some large positive constant.*

Proof: Recall, $A_t = \hat{\mathbf{P}}_t(\hat{\Sigma}_{t-1} + \lambda\mathbf{I})\hat{\mathbf{P}}_t$ and $B_t = \hat{\mathbf{U}}_t^\top(\hat{\Sigma}_{t-1} + \lambda\mathbf{I})\hat{\mathbf{U}}_t$. We can see that $\hat{\mathbf{U}}_t B_t \hat{\mathbf{U}}_t^\top = A_t$.

$$\begin{aligned}
2\sum_{t=t_b}^{T} \beta_{t,\delta} \|\hat{\mathbf{P}}_t X_t\|_{A_t^\dagger} &\leq 2\beta_{T,\delta} \sum_{t=t_b}^{T} \|\hat{\mathbf{P}}_t X_t\|_{A_t^\dagger} \\
&\leq 2\beta_{T,\delta}\left(\frac{L}{\sqrt{\lambda}}\left(1 + \frac{2}{p}\right)\sum_{t=t_b}^{T}\epsilon_t + \sum_{t=t_b}^{T}\|\hat{\mathbf{U}}_t^\top \hat{X}_t\|_{B_t^{-1}}\right) \\
&\qquad\qquad \text{(From Lemma 22)} \\
&\leq 2\beta_{T,\delta}\left(L\sqrt{\frac{\zeta}{\lambda}}\left(1 + \frac{2}{p}\right)(2\sqrt{T} - 2\sqrt{t_b + 1} + 1)\right. \\
&\leq 2\beta_{T,\delta}\left(LC\sqrt{\frac{\zeta T}{\lambda}}\left(1 + \frac{2}{p}\right) + \sum_{t=t_b}^{T}\|\hat{\mathbf{U}}_t^\top \hat{X}_t\|_{B_t^{-1}}\right) \\
&\qquad\qquad \text{(From Equation (4.35))} \\
&\leq 2\beta_{T,\delta}\left(LC\sqrt{\frac{\zeta T}{\lambda}}\left(1 + \frac{2}{p}\right) + \sqrt{T - t_b + 1}\sqrt{\sum_{t=t_b}^{T}\|\hat{\mathbf{U}}_t^\top \hat{X}_t\|_{B_t^{-1}}^2}\right) \\
&\qquad\qquad \text{(Using Cauchy Schwarz)} \\
&\leq 2\beta_{T,\delta}\left(LC\sqrt{\frac{\zeta T}{\lambda}}\left(1 + \frac{2}{p}\right) + \sqrt{T}\sqrt{\sum_{t=1}^{T}\|\hat{\mathbf{U}}_t^\top \hat{X}_t\|_{B_t^{-1}}^2}\right).
\end{aligned}$$

Using Lemma 23, we can further simplify the bound on term 3 as:

$$2\sum_{t=t_b}^{T}\beta_{t,\delta}\|\hat{\mathbf{P}}_t X_t\|_{A_t^\dagger} \leq 2\beta_{T,\delta}\left(LC\sqrt{\frac{\zeta T}{\lambda}}\left(1+\frac{2}{p}\right)\right.$$
$$\left.+\sqrt{T}\sqrt{\left(1+\frac{2}{p}\right)^2\frac{L^2\zeta^2}{\lambda}+\frac{\zeta L^2}{\lambda}\left(1+2\log(T+1)\right)+4m\log\mathcal{L}}\right)$$

where, $\mathcal{L}=\log\left(1+\frac{12\zeta L^2(1+\log t)}{m\lambda}\left(1+\frac{2}{p}\right)^2+\frac{tL^2}{m\lambda}\right)$. Recall that from Lemma 23, $\lambda > 12\zeta L^2\left(1+\frac{2}{p}\right)^2+L^2)$. Using this we get:

$$2\sum_{t=1}^{T}\beta_{t,\delta}\|\hat{\mathbf{P}}_t X_t\|_{A_t^\dagger} \leq C\beta_{T,\delta}\left(L\sqrt{\zeta T}+\sqrt{(1+\zeta+2\log(T+1)+4m\log\mathcal{L})T}\right) \quad (4.38)$$

■

Lemma 22 *At any round t, we have defined* $A_t=\hat{\mathbf{P}}_t(\hat{\Sigma}_{t-1}+\lambda\mathbf{I})\hat{\mathbf{P}}_t$ *where,* $\hat{\mathbf{P}}_t=\hat{\mathbf{U}}_t\hat{\mathbf{U}}_t^\top$ *and* $\hat{\Sigma}_t=\sum_{i=1}^{t}\hat{X}_i\hat{X}_i^\top$. *Also,* $B_t=\hat{\mathbf{U}}_t^\top(\hat{\Sigma}_{t-1}+\lambda\mathbf{I})\hat{\mathbf{U}}_t$. *Then,*

$$\|\hat{\mathbf{P}}_t X_t\|_{A_t^\dagger} \leq \frac{L}{\sqrt{\lambda}}\left(1+\frac{2}{p}\right)\epsilon_t + \|\hat{\mathbf{U}}_t^\top\hat{X}_t\|_{B_t^{-1}}.$$

Proof:

$$\begin{aligned}
\|\hat{\mathbf{P}}_t X_t\|_{A_t^\dagger} &= \|\hat{\mathbf{P}}_t(X_t-\hat{X}_t+\hat{X}_t)\|_{A_t^\dagger} \\
&\leq \|\hat{\mathbf{P}}_t(X_t-\hat{X}_t)\|_{A_t^\dagger} + \|\hat{\mathbf{P}}_t\hat{X}_t\|_{A_t^\dagger} \\
&\leq \|(A_t^\dagger)^{1/2}\hat{\mathbf{P}}_t(X_t-\hat{X}_t)\|_2 + \|\hat{\mathbf{P}}_t\hat{X}_t\|_{A_t^\dagger} \\
&\leq \|(A_t^\dagger)^{1/2}\|_2\|\hat{\mathbf{P}}_t\|_2\|(X_t-\hat{X}_t)\|_2 + \|\hat{\mathbf{P}}_t\hat{X}_t\|_{A_t^\dagger} \\
&\leq \frac{L}{\sqrt{\lambda}}\left(1+\frac{2}{p}\right)\epsilon_t + \|\hat{\mathbf{P}}_t\hat{X}_t\|_{A_t^\dagger} && \text{(Using Lemma 5)} \\
&= \frac{L}{\sqrt{\lambda}}\left(1+\frac{2}{p}\right)\epsilon_t + \|\hat{\mathbf{U}}_t^\top\hat{X}_t\|_{B_t^{-1}} && (\hat{X}_t^\top\hat{\mathbf{U}}_t B_t^{-1}\hat{\mathbf{U}}_t^\top\hat{X}_t = \hat{X}_t^\top\hat{\mathbf{P}}_t A_t^\dagger\hat{\mathbf{P}}_t\hat{X}_t)
\end{aligned}$$

■

Lemma 23 *Let $\hat{X}_t$ denote the imputed vector at time t, and $B_t = \hat{\mathbf{U}}_t^\top(\hat{\Sigma}_{t-1} + \lambda\mathbf{I})\hat{\mathbf{U}}_t$, where $\hat{\Sigma}_{t-1}$ is the covariance matrix constructed from the imputed action vectors chosen up to time $t-1$. Then, with probability at least $1-3\delta$, the following holds for all $t \in [t_b, T]$:*

$$\sum_{t=t_b}^{T} \|\hat{\mathbf{U}}_t^\top \hat{X}_t\|^2_{B_t^{-1}} \leq \left(1+\frac{2}{p}\right)^2 \frac{L^2\zeta^2}{\lambda} + \frac{\zeta L^2}{\lambda}\left(1+2\log(T+1)\right) + 4m\log\left(1+\frac{\zeta L^2(1+\log t)}{m\lambda}\left(1+\frac{2}{p}\right)^2 + \frac{tL^2}{m\lambda}\right).$$

where, $\zeta := \frac{12m\lambda_1^2 M^2}{p^2\lambda_m^2 K}\log\left(\frac{2dT}{\delta}\right)$.

Proof: Let us recall the definitions of certain terms:

- $A_t = \hat{\mathbf{P}}_t(\hat{\Sigma}_{t-1} + \lambda\mathbf{I})\hat{\mathbf{P}}_t$ where, $\hat{\mathbf{P}}_t = \hat{\mathbf{U}}_t\hat{\mathbf{U}}_t^\top$ and $\hat{\Sigma}_t = \sum_{i=1}^{t}\hat{X}_i\hat{X}_i^\top$
- $B_t = \hat{\mathbf{U}}_t^\top(\hat{\Sigma}_{t-1} + \lambda\mathbf{I})\hat{\mathbf{U}}_t$.
- We define the following quantity: $\zeta := \frac{12m\lambda_1^2 M^2}{p^2\lambda_m^2 B}\log\left(\frac{2dT}{\delta}\right)$

$$\begin{aligned}
\|\hat{\mathbf{U}}_t^\top \hat{X}_t\|^2_{B_t^{-1}} &= \|(\hat{\mathbf{U}}_t - \hat{\mathbf{U}}_{t+1} + \hat{\mathbf{U}}_{t+1})^\top \hat{X}_t\|^2_{B_t^{-1}} \\
&\leq \|(\hat{\mathbf{U}}_t - \hat{\mathbf{U}}_{t+1})^\top \hat{X}_t\|^2_{B_t^{-1}} + \|\hat{\mathbf{U}}_{t+1}^\top \hat{X}_t\|^2_{B_t^{-1}} \\
&\leq \underbrace{\frac{\|\hat{X}_t\|_2^2}{\lambda}(\|\hat{\mathbf{U}}_t - \mathbf{U}\|_2^2 + \|\mathbf{U} - \hat{\mathbf{U}}_{t+1}\|_2^2)}_{\text{Goes down } \propto 1/t} + \|\hat{\mathbf{U}}_{t+1}^\top \hat{X}_t\|^2_{B_t^{-1}}
\end{aligned}$$

Summing from t_b to T we get:

$$\begin{aligned}
\sum_{t=t_b}^{T} \|\hat{\mathbf{U}}_t^\top \hat{X}_t\|^2_{B_t^{-1}} &\leq \sum_{t=1}^{T} \|\hat{\mathbf{U}}_t^\top \hat{X}_t\|^2_{B_t^{-1}} \\
&\leq \underbrace{\sum_{t=1}^{T} \frac{\|\hat{X}_t\|_2^2}{\lambda}(\|\hat{\mathbf{U}}_t - \mathbf{U}\|_2^2 + \|\mathbf{U} - \hat{\mathbf{U}}_{t+1}\|_2^2)}_{\text{T1}} + \underbrace{\sum_{t=1}^{T} \|\hat{\mathbf{U}}_{t+1}^\top \hat{X}_t\|^2_{B_t^{-1}}}_{\text{T2}}
\end{aligned}$$

Bounding T1:

$$\begin{aligned}
\sum_{t=1}^{T} \frac{\|\hat{X}_t\|_2^2}{\lambda}(\|\hat{\mathbf{U}}_t - \mathbf{U}\|_2^2 + \|\mathbf{U} - \hat{\mathbf{U}}_{t+1}\|_2^2) &\leq \sum_{t=1}^{T} \frac{\|\hat{X}_t - X_t\|_2^2 + \|X_t\|_2^2}{\lambda} \left(\epsilon_t^2 + \epsilon_{t+1}^2\right) \\
&\leq \sum_{t=1}^{T} \frac{\|\hat{X}_t - X_t\|_2^2 + \|X_t\|_2^2}{\lambda} \left(\frac{\zeta}{t} + \frac{\zeta}{t+1}\right) \\
&\leq \sum_{t=1}^{T} \left(12\left(1+\frac{2}{p}\right)^2 \frac{L^2\zeta}{\lambda t} + \frac{L^2}{\lambda}\right)\left(\frac{12\zeta}{t} + \frac{12\zeta}{t+1}\right) \\
&\leq \left(1+\frac{2}{p}\right)^2 \frac{L^2\zeta^2}{\lambda} + \frac{\zeta L^2}{\lambda}\left(1 + \log T + \log(T+1)\right) \\
&\leq \left(1+\frac{2}{p}\right)^2 \frac{L^2\zeta^2}{\lambda} + \frac{\zeta L^2}{\lambda}\left(1 + 2\log(T+1)\right)
\end{aligned}$$

Bounding T2:

$$\begin{aligned}
\det(B_{T+1}) &= \det(B_T + \hat{\mathbf{U}}_{T+1}\hat{X}_T\hat{X}_T\hat{\mathbf{U}}_{T+1}) \\
&= \det(B_T^{1/2}(\mathbf{I}_m + B_T^{-1/2}\hat{\mathbf{U}}_{T+1}\hat{X}_T\hat{X}_T^\top\hat{\mathbf{U}}_{T+1}B_T^{-1/2})B_T^{1/2}) \\
&= \det(B_T)(1 + \|\hat{\mathbf{U}}_{T+1}\hat{X}_T\|_{B_T^{-1}}^2) \\
&= \lambda^m \prod_{t=1}^{T}(1 + \|\hat{\mathbf{U}}_{t+1}\hat{X}_t\|_{B_t^{-1}}^2).
\end{aligned}$$

Taking natural log on both sides:

$$\sum_{t=1}^{T} \log(1 + \|\hat{\mathbf{U}}_{t+1}\hat{X}_t\|_{B_{t-1}^{-1}}^2) = \log\left(\frac{\det(B_{T+1})}{\lambda^m}\right) \tag{4.39}$$

$\det(B_{T+1}) = \det(\hat{\mathbf{U}}_{T+1}^\top\hat{\Sigma}_T\hat{\mathbf{U}}_{T+1} + \lambda I_m) = \alpha_1\alpha_2\ldots\alpha_m$, where α_i's are the eigenvalues

of $\dot{B}_T$. Using the AM-GM inequality, $\sqrt[m]{\alpha_1\alpha_2\ldots\alpha_m} \leq \frac{1}{m}\sum_{i=1}^{m}\alpha_i$, we get:

$$\begin{aligned}
\alpha_1\alpha_2\ldots\alpha_m &\leq \left(\frac{1}{m}\sum_{i=1}^{m}\alpha_i\right)^m \\
&= \left(\lambda + \frac{1}{m}\sum_{t=1}^{T}\text{tr }(\hat{\mathbf{U}}_{T+1}^\top \hat{X}_t\hat{X}_t^\top\hat{\mathbf{U}}_{T+1})\right)^m \\
&= \left(\lambda + \frac{1}{m}\sum_{t=1}^{T}\text{tr }(\hat{X}_t^\top\hat{\mathbf{U}}_{T+1}\hat{\mathbf{U}}_{T+1}^\top \hat{X}_t)\right)^m \qquad \text{(Cyclic property of trace)} \\
&= \left(\lambda + \frac{1}{m}\sum_{t=1}^{T}\text{tr }(\hat{X}_t^\top\hat{\mathbf{P}}_{T+1}\hat{X}_t)\right)^m \\
&\leq \left(\lambda + \frac{1}{m}\sum_{t=1}^{T}\|\hat{X}_t\|_2^2\right)^m \\
&\leq \left(\lambda + \frac{1}{m}\sum_{t=1}^{T}(\|\hat{X}_t - X_t\|_2^2 + L^2)\right)^m \qquad \text{(Using triangle inequality)} \\
&\leq \left(\lambda + \frac{\zeta L^2(1+\log t)}{m}\left(1+\frac{2}{p}\right)^2 + \frac{tL^2}{m}\right)^m
\end{aligned}$$

Substituting Equation (4.39) in the above expression we get:

$$\begin{aligned}
\sum_{t=1}^{T}\log(1+\|\hat{\mathbf{U}}_{t+1}\hat{X}_t\|_{B_t^{-1}}^2) &= \log\left(\frac{\det(B_{T+1})}{\lambda^m}\right) \\
&\leq m\log\left(1+\frac{\zeta L^2(1+\log t)}{m\lambda}\left(1+\frac{2}{p}\right)^2 + \frac{tL^2}{m\lambda}\right).
\end{aligned}$$

Here,

$$\begin{aligned}
\|\hat{\mathbf{U}}_{t+1}^\top\hat{X}_t\|_{B_t^{-1}}^2 &\leq \|B_t^{-1}\|_2\|\hat{X}_t\|^2 \qquad \text{(Using Cauchy-Schwarz inequality)} \\
&\leq \frac{1}{\lambda}\left(\|\hat{X}_t - X_t\|_2^2 + \|X_t\|_2^2\right) \\
&\leq \frac{1}{\lambda}\left(\frac{\zeta L^2}{t}\left(1+\frac{2}{p}\right)^2 + L^2\right).
\end{aligned}$$

For $\lambda \geq \max\left\{1, \zeta L^2\left(1+\frac{2}{p}\right)^2 + L^2\right\}$, the quantity $\|\hat{\mathbf{U}}_t^\top\hat{X}_t\|_{B_t^{-1}}^2 \leq 1$. Using the fact

when $x \in [0,1]$, $x \leq 4\log(1+x)$ holds true, we get:

$$\begin{aligned}\sum_{t=1}^{T} \|\hat{\mathbf{U}}_{t+1}^{\top}\hat{X}_t\|_{B_t^{-1}}^2 &\leq 4\sum_{t=1}^{T}\log(1+\|\hat{\mathbf{U}}_{t+1}\hat{X}_t\|_{B_t^{-1}}^2) \\ &\leq 4m\log\left(1+\frac{\zeta L^2(1+\log t)}{m\lambda}\left(1+\frac{2}{p}\right)^2+\frac{tL^2}{m\lambda}\right)\end{aligned}$$

Here, the last inequality follows from using Equation (Using Cauchy-Schwarz inequality).

■

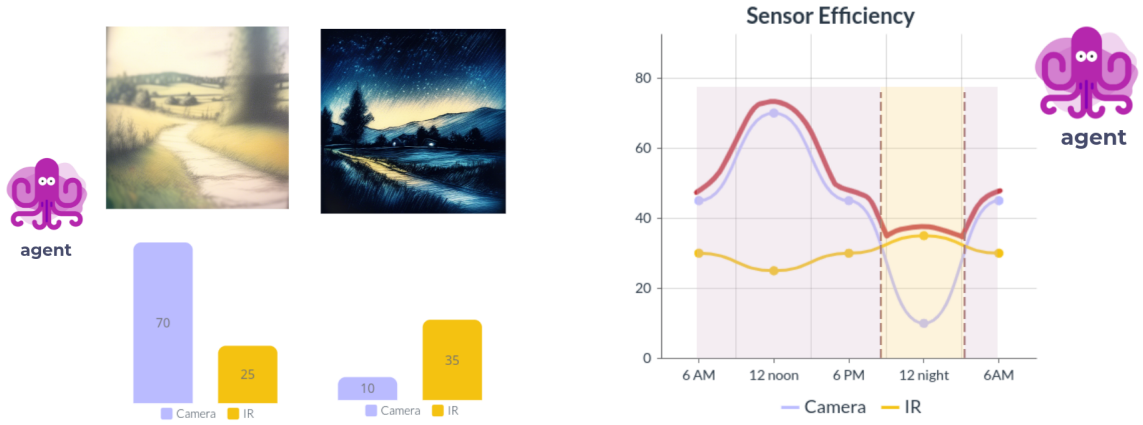
Chapter 5

PERIODIC BANDITS WITH LINEAR REWARDS

5.1 Introduction

Sequential decision-making under uncertainty is crucial in a wide variety of fields. Ideally, given ample time, one would exhaustively sample all available options before making decisions. However, in modern problems which present the decision maker with an enormous number of choices, such an approach is infeasible. The *Multi-armed bandits* (MAB) framework Slivkins et al. 2019 addresses this by efficiently identifying optimal options in minimal time. Central to MABs is the Exploration-exploitation dilemma: one must balance exploring unknown choices and exploiting the best-known option. Given its strong theoretical foundations and its efficacy in a wide range of domains like recommendation systems, clinical trials, and online advertising, this framework and its variants have received much attention in recent years Slivkins et al. 2019; Langford et al. 2007; Abbasi-yadkori et al. 2011.

A key limitation of this framework, however, is its traditional reliance on stationarity of the underlying "reward" distribution. Real-world applications often exhibit non-stationarity. Introducing non-stationary reward distributions complicates matters due to potential erratic patterns. Although there have been endeavors to address this (see, e.g., Besbes et al. 2019; Garivier et al. 2008; Slivkins et al. 2008), formulating a universal learning policy for non-stationarity remains challenging. To motivate this further consider the example in Figure 19, where the task is to select the most suitable sensor for viewing a scene. Choosing between a camera and an IR sensor, for instance, does not have a straightforward answer—it depends on factors like the time



(a) (b)

Figure 19. (a) Illustrations of performance efficiency for Camera and IR sensor as time of day, (b) Allocation of energy to different sensors for optimal image capture performance.

of day. Selecting the optimal sensor based on the time of day can thus be modeled as a periodic process.

With the explosion in the possibility of choices, this approach hardly seems like a practical strategy. *Multi-armed bandits* (MABs) is a framework that excels at efficiently choosing the best option within as little time as possible. Termed as Exploration-exploitation dilemma in the MAB literature, a MAB decision-maker is repeatedly offered a set of choices (arms) with unknown rewards to choose from. In order to perform well in this framework, one needs to strike a balance between trying out an unknown choice (exploration) and choosing the best option seen so far (exploitation). There are several popular algorithms and variants of MABs, including semi-uniform strategies Langford et al. 2007, contextual bandits Abbasi-yadkori et al. 2011, and dueling bandits Yue et al. 2012. The research on MABs continues to expand with applications in recommendation systems, clinical trials, online advertising, and more. While the results showcased in the literature are certainly impressive, one central

limitation of the traditional MABs is the stationarity assumption on the reward distribution.

Most practical applications in real-world scenarios have some kind of non-stationarity associated with them. Allowing non-stationarity in reward distribution adds a plethora of challenges due to the bizarre non-stationary patterns it can possibly follow. Despite the practical relevance and several attempts Besbes et al. 2019; Besson et al. 2019; Garivier et al. 2008; Gupta et al. 2011; Slivkins et al. 2008, it is difficult to develop a generic learning policy for non-stationary rewards, especially when the dynamics can permitted to change arbitrarily over time.

In this work, we focus on *Periodic Bandits*, a class of non-stationary bandits that are characterized by a periodic pattern in their rewards. Such periodicity is common in a range of real-world scenarios, such as cell-tower congestion, advertisement trends, and behavior of electronic systems reliant on discharging power sources. Ignoring these patterns can result in highly suboptimal decisions Villamediana et al. 2019. Incorporating periodicity into multi-armed bandit algorithms enables one to make decisions that align more closely with the natural rhythms and temporal variations present in the problem domain.

Research such as Benedetto et al. 2020 has addressed seasonal reward shifts, while Re et al. 2021 leverages historical data for sudden changes. Other studies, like Zhou et al. 2021, focus on regime-switching rewards, while Q. Chen et al. 2023 considers rewards based on auto-regressive models. Hengrui Cai et al. 2021 integrates periodicity in Gaussian process bandits. Our work aligns most closely with N. Chen et al. 2020, which combines Fourier analysis with a confidence-bound-based learning procedure to learn the periods and minimize the regret.

In Benedetto et al. 2020, the authors consider seasonal changes in reward functions

and present a contextual bandit algorithm to detect and adapt to abruptly changing environments. In Re et al. 2021, the authors exploit historic data to integrate past information in the abruptly changing nonstationary setting. In Zhou et al. 2021, the authors focus on rewards exhibiting regime switching. Specifically, the distributions of the random rewards generated are modulated by a common underlying state modeled as a finite-state Markov chain. The authors of Q. Chen et al. 2023, consider a stochastic temporal structure, where the expected reward of each arm is governed by an auto-regressive (AR) model. In another paper Hengrui Cai et al. 2021, the focus is on integrating periodicity in Gaussian process bandits. Our work is more closely related to N. Chen et al. 2020 where, with the non-stationary, periodic evolution of the environment, the authors propose a two-stage policy combining Fourier analysis with a confidence-bound-based learning procedure to learn the periods and minimize the regret.

This paper proposes a tractable methodology for tacking the periodic bandit framework. To this end, we utilize the framework of Ramanujan Periodicity Transforms (RPT) to estimate the length of the period and identify the fundamental periods if the signal is a combination of two or more periodic signals. The authors in Tenneti et al. 2015; Vaidyanathan 2014 introduced the notion of RPT and showed that one can utilize RPTs to estimate the underlying period of a periodic signal. In addition, the authors demonstrated that RPT-based methods are more robust in the presence of noise and showed the advantages of RPTs over the classical DFT-based techniques Tenneti et al. 2015. RPTs have been used in practice such as detecting periodicity in visually evoked potentials in brain-computer interfaces Saidi et al. 2019 and detecting the tandem DNA repeats Tenneti et al. 2016 and have shown promising results.

Contributions. The main contributions of this work are the following.

a) We propose an online learning algorithm called Bandit Tracking System via Ramanujan Periodic transform (BTS-RaP) for non-stationary environments with seasonal patterns and unknown periods.

b) We propose the use of RPT dictionaries to estimate length of periods across different arms which are known to overcome the limitations of DFT-based technique.

c) Using computer simulations we show that BTS-RaP algorithm can achieve sublinear regret.

5.2 Preliminaries

5.2.1 Ramanujan Periodicity Transforms

In this section, we briefly review the structure of the RPT dictionary, and their applicability to estimate the period of a periodic signal.

5.2.1.1 RPT dictionaries

RPT dictionaries are constructed based on the notion of Ramanujan sums, defined as Ramanujan 1918

$$c_p(n) = \sum_{\substack{k=1 \\ (k,p)=1}}^{p} \exp(j2\pi kn/p), \tag{5.1}$$

where (k,p) is the greatest common divisor (gcd) of k and p. We use $\mathbf{c}_p$ to show the vector form of $c_p(n)$, and $\mathbf{c}_p^{(i)}$ to show the circularly shifted version of $\mathbf{c}_p$ with step size i. Then, for each value p one can construct a $p \times \phi(p)$ submatrix $\mathbf{C}_p$ as follows

$$\mathbf{C}_p = \begin{bmatrix} \mathbf{c}_p & \mathbf{c}_p^{(1)} & \dots & \mathbf{c}_p^{(\phi(p)-1)} \end{bmatrix}, \tag{5.2}$$

where $\phi(p)$ is the Euler totient function, that is the number of integers that are co-prime to p. Vaidyanathan in Vaidyanathan 2014 has showed that the $\phi(p)$ columns

in $\mathbf{C}_p$ are linearly independent. One can construct the RPT dictionary $\mathbf{K}$ in three steps. First, building all the submatrices $\mathbf{C}_p$ for every $p \in \mathbb{P}$, where $\mathbb{P} = \{1, 2, \ldots, P_{\max}\}$ and $P_{\max}$ is the largest possible period in the signal. Second, building the $L \times \phi(p)$ submatrices $\mathbf{R}_p$, by periodically extending all the columns of $\mathbf{C}_p$ to length L. Third, concatenating the matrices $\mathbf{R}_p$ as

$$\mathbf{K} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_2 & \ldots & \mathbf{R}_{P_{\max}} \end{bmatrix} . \tag{5.3}$$

5.2.1.2 Period estimation using RPT dictionary

Discrete periodic signals can be expressed using the RPT dictionary in a noise-free setup as

$$\mathbf{y} = \mathbf{K}\mathbf{x} \tag{5.4}$$

where $\mathbf{y}$ is the vector form of the periodic signal $y(n)$ with period p, $\mathbf{K}$ is the RPT dictionary introduced in section 5.2.1, and $\mathbf{x}$ is the sparse representation of the periodic signal under the RPT dictionary. Given a sufficiently long vector $\mathbf{y}$, vector $\mathbf{x}$ exhibits a sparse structure and its non-zero values correspond to the submatrices in $\mathbf{K}$, that have periodic columns with periods q_i that are divisors of p, or $q_i | p$. Therefore, it is possible to estimate the period of a periodic signal by first recovering the sparse representation of the signal under the RPT dictionary. Then, the support set of the signal identifies the divisors of the underlying period of the signal. The support set of a sparse vector are a set of indices that contain the location of the non-zero values of the vector. Finally, the estimate of the period is equal to the least common multiplier (LCM) of the divisors from the recovered support set. One can recover the support of the sparse vector $\mathbf{x}$ using sparse recovery algorithms Tropp 2004; Baraniuk 2007. In this work, we adopt the proposed approach in Tenneti et al. 2015 and solve the following minimization program:

$$\min \|\mathbf{Dx}\|_p \quad \text{s.t.} \quad \mathbf{y} = \mathbf{Kx} \tag{5.5}$$

where, $\mathbf{D}$ is a diagonal penalty matrix. The i-th entry on the diagonal is equal to p_i^2, where p_i is the period of the i-th column of the dictionary $\mathbf{K}$. Here, p can be chosen to be 1 or 2 based on which optimization problem we wish to solve. We show the recovery of support of the sparse vector $\mathbf{x}$ under both optimization problems.

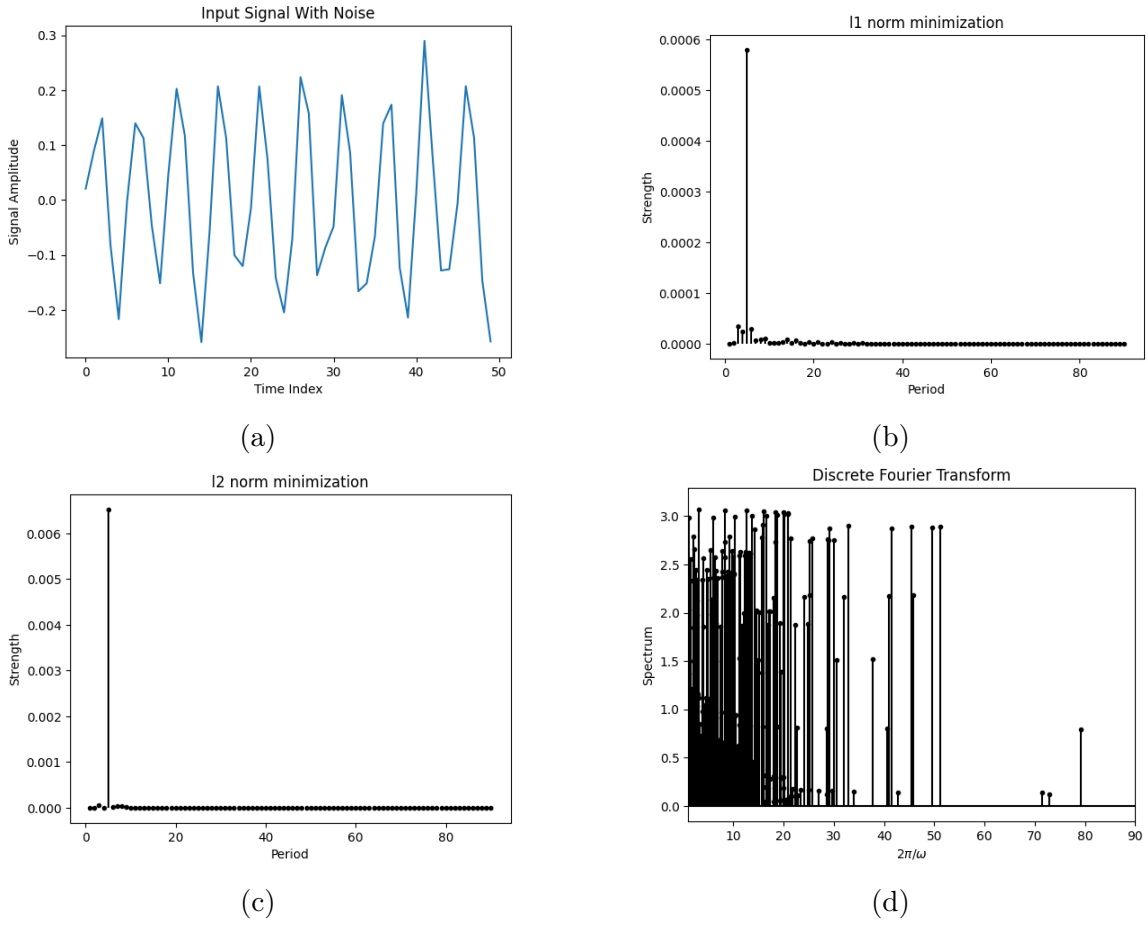


Figure 20. (a) A period-5 sinusoidal signal with noise (b) , (c) - The strength vs period plots for the solutions of the convex program (5.5) using Ramanujan dictionaries (d) Recovery of period based on DFT dictionary.

5.3 Problem Setup

Consider a multi-armed bandit setting with $\mathcal{K}$ being the set of all arms such that mean of each arm $i \in \mathcal{K}$ is represented by function $\mu_i : \mathbb{N} \to [a, b] \quad \forall i \in [K]$ such that $\mu_i[t + T_i] = \mu_i[t]$ for some unknown $T_i \in \mathbb{N}$. Throughout the paper, we sometimes refer $\mu_i[t]$ as $\mu_{t,i}$. At each round, the learner chooses an arm $a_t \in \mathcal{K}$ to sample and observes a noisy reward

$$r_{t,i} = \mu_{t,i} + \eta_{t,i}\,,$$

where, $\{\eta_{i,t}\}_{i,t}$ are i.i.d. noise samples from a σ^2- sub-Gaussian distribution. The goal of the problem is to minimize the regret up to a known time horizon T defined as,

$$\mathcal{R}(T) = \sum_{t=0}^{T} \left(\max_{i \in \mathcal{K}} \mu_{t,i} - \mu_{t,a_t} \right)\,, \tag{5.6}$$

where, the decision maker chooses an arm a_t at time step t. The aim of the work is to propose an algorithm to minimize regret as mentioned in (5.6). This can only be obtained if the decision maker chooses an arm that is optimal *for every time* t. If, at the current time instant, one of the arms is optimal, it is not necessary that the previously chosen arm will be optimal again at the next time step, which is well-suited for handling time-varying reward changes. The notion in (5.6) is different than the standard notion of regret Bubeck et al. 2015, which focuses on selecting the one optimal choice for every time-step t. In order the necessitate the requirement of such a notion, please consider the following toy example.

Toy Example 1 : Consider the setup for two arms with mean values as a function of time as follows: $\mu_1(t) = sin\,(t)$ and $\mu_2(t) = cos\,(t)$. The reward from the arms is as follows:

$$r(t, i) = \mu_i(t) + \eta_t \tag{5.7}$$

where η_t is Gaussian independent noise. Here the correct choice of arms to minimize regret can be given by the following mapping,

$$\mu^*(t) = \begin{cases} \mu_2(t) & t \in [2n\pi - \frac{3\pi}{4}, 2n\pi + \frac{\pi}{4}] \\ \mu_1(t) & t \in [2n\pi + \frac{\pi}{4}, 2n\pi + \frac{5\pi}{4}] \end{cases} \tag{5.8}$$

Note that the following is true:

$$\begin{aligned} \mathbb{E}_t[\mu_1(t)] &= \lim_{T\to\infty} \frac{\int_{t=0}^{T} \mu_1(t)}{T} = 0, \\ \mathbb{E}_t[\mu_2(t)] &= \lim_{T\to\infty} \frac{\int_{t=0}^{T} \mu_2(t)}{T} = 0 \end{aligned} \tag{5.9}$$

Note that $\max_{i=1,2} \mathbb{E}_t[\mu_i] = \mathbb{E}_t[\mu_1] = \mathbb{E}_t[\mu_2] = 0$ and hence the standard regret can be computed to be $\mathcal{R}_{\text{standard}}(T) = \max_{i=1,2} \int_{t=0}^{T} \mathbb{E}[\mu_i] - \mathbb{E}[\mu_{x_t}] = 0$ for any choice of arm 1 or arm 2 (provided we define $\mu^* = \frac{\int_{t=0}^{\pi} \mu^*(t)}{\pi}$). As per our definition of regret (5.6), any choice of the arm, whether 1 or 2, the regret can be computed to be

$$R(n\pi) = n\pi \quad n \in \mathbb{N} \tag{5.10}$$

which is expected regret since, the optimal arm choice for minimum regret should switch at $t = 2n\pi + \frac{\pi}{4}, 2n\pi + \frac{5\pi}{4}$. Hence we can showcase that the standard notion of bandit is insufficient for tracking progress for Non-stationary bandits with periodicity.

5.3.1 Baseline method

Recently N. Chen et al. 2020 proposed for addressing periodic bandits a two-stage approach which provides a sub-linear regret that scales as $\mathcal{O}\left(\sqrt{T \sum_{i=1}^{n} T_i}\right)$, where T_i is the period of arm i. The authors first propose to use (DFT) to estimate the length of the periods T_i's. Since the mean of arm i returns to the same value every T_i steps, the authors propose that for every arm i, the number of 'effective arms' is T_i (1 arm for every step until time reaches T_i). Therefore, we end up with $\hat{d} := \sum_k \hat{T}_k$

effective arms (unique mean rewards) to learn. In the second stage of the algorithm, the authors utilize the estimated number of effective arms to implement UCB-based approach to minimize regret (5.6). We refer this as MAB-UCB.

This approach suffers drawbacks due to the utilization of DFT as well as two distinct stages leading to sample inefficiency. We address this by using RPT and merging the two stages into one main algorithm thereby painting optimal sample efficiency.

5.4 Proposed Approach: BTS-RaP (Bandit Tracking System)

We first provide an overview of the linear bandits and then show how it connects to RPT-based reward representation.

5.4.1 Linear bandits

Linear bandits Abbasi-yadkori et al. 2011 have emerged as a powerful and versatile tool in the field of bandit research literature. These algorithms are particularly well-suited for scenarios where the relationship between actions and rewards can be approximated linearly. Let the arm set be defined by the set $\mathcal{K}$. In a linear bandit setup, every arm is associated with a feature vector $\mathbb{R}^d$ such that $d < n$. On sampling the arm i at time t, the reward observed satisfies the relation $r_{t,i} = \langle \mathbf{a}_i, \boldsymbol{\theta}^* \rangle + \eta_t$, where $\boldsymbol{\theta}^* \in \mathbb{R}^d$ dented the unknown reward parameter, $\mathbf{a}_i \in \mathbb{R}^d$ denote the feature vector associated with arm i and η_t is the i.i.d. Gaussian noise realized from a σ^2-subgaussian distribution at time t.

Due to the low dimensional structure of the linear bandit problem, it has been proven, both theoretically and experimentally that the regret upper bound scales as $\mathcal{O}(\sqrt{dT})$ (sublinear in time T), where d is the feature dimensionality. Note that for

the case of stationary linear bandit, the regret takes the form as defined below,

$$\mathcal{R}_{LB}(T) = \sum_{t=0}^{T} \left(\max_{k \in \mathcal{K}} \langle \mathbf{a}_k, \boldsymbol{\theta}^* \rangle - \langle \mathbf{a}_t, \boldsymbol{\theta}^* \rangle \right) \tag{5.11}$$

Taking a step further Y. Wang et al. 2020 showcases that the regret upper bound can be further tightened to nearly $\mathcal{O}(\sqrt{s_0 T})$, where $s_0 : \|\boldsymbol{\theta}^*\|_0 \leq s_0$ is the support of $\boldsymbol{\theta}^*$.

5.4.2 Connection to RPT decomposition

Let $\mathbf{K}$ be the RPT dictionary and $\mathbf{x}_i$ be the corresponding sparse vector associated with the arm i with support set S_i. We can map our problem setup to linear bandits as follows:

i) Construct the block diagonal matrix $\mathbf{K}_{\mathcal{K}} = \text{diag}(\mathbf{K}, \mathbf{K}, \dots)$, where each $\mathbf{K} \in \mathbb{R}^{T \times \phi(P_{\max})}$ are blocks on the diagonal and constructed as per Equation (5.3). For an arm i, the arm features, at time t is $\mathbf{a}_{t,i} = \mathbf{K}_{\mathcal{K}}[i * T + t]$, which is the $(i * T + t)^{th}$ row of $\mathbf{K}_{\mathcal{K}}$.

ii) The unknown feature vector $\boldsymbol{\theta}^*$ is a vector stack of $\mathbf{x}_i$, where $\mathbf{x}_i$ is the true solution to the minimization problem (5.5) in the noiseless case. The reward is obtained as $r_{t,i} = \langle \mathbf{a}_{t,i}, \mathbf{x}_i \rangle$.The pseudocode of the proposed algorithm is provided in Algorithm 5 Following directly from Y. Wang et al. 2020, we can provide the following theoretical backing to BTS-RaP:

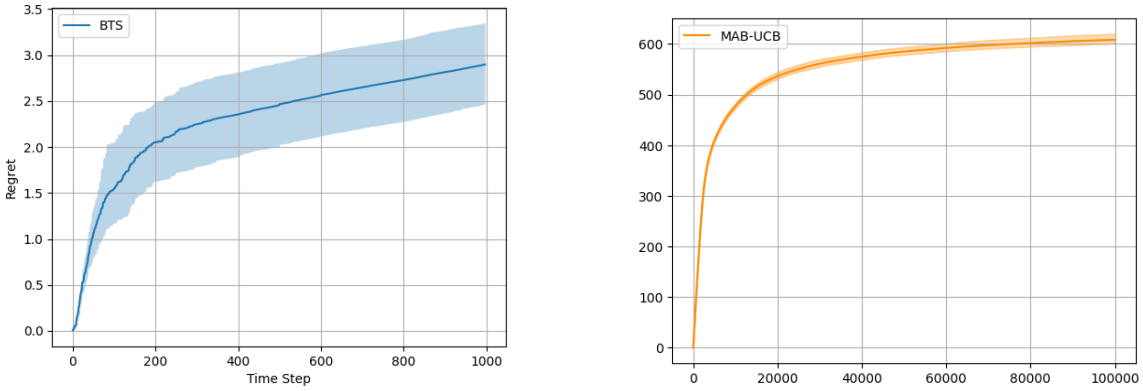
Theorem 8 *Let $\mathbf{x}_i$ be the sparse representation of a periodic signal under the RPT dictionary with support set $\mathcal{S}_i$ that has $|\mathcal{S}_i|$ nonzero values, for all $\mathbf{x}_i$ $i \in \mathcal{K}$, then the regret of Bandit Tracking System (BTS-RaP) is upper bounded by $\mathcal{O}(\sqrt{T \sum_{i \in \mathcal{K}} |\mathcal{S}_i|})$.*

Algorithm 5 Bandit Tracking System (BTS-RaP)

Given T, K arms, form $\mathbf{K} \in \mathbb{R}^{T \times \phi(P_{\max})}$ pull each arm once and form the observation vector $\boldsymbol{\mu}_i \ \forall i \in \mathcal{K}$
Create mini-dictionaries $\mathbf{K}_i$ for $i \in \mathcal{K}$ which will grow as arms get pulled. Initially each $\mathbf{K}_i$ is of size $1 \times \phi(P_{\max})$
Initialize support for each arm $\mathbf{x}_i$ for all $i \in \mathcal{K}$
for $t = K+1 \dots T$ **do**
    Choose arm $a_t = \arg\max_{k \in \{1,\dots,K\}} \langle \mathbf{K}[t], \mathbf{x}_i \rangle + \sqrt{\frac{2\,\alpha\,\ln t}{n_{t-1,i}}}$, where, $\mathbf{K}[t]$ is the t^{th} row of $\mathbf{K}$
    Append the row $\mathbf{K}[t]$ to the mini-dictionary $\mathbf{K}_{a_t}$
    Observe $r_{t,a_t} = \mu_{t,a_t} + \eta_t$ and append this observation to $\boldsymbol{\mu}_{a_t}$
    Solve : $\min \|\mathbf{D}\mathbf{x}_{a_t}\|_2 \quad \text{s.t.} \quad \boldsymbol{\mu}_{a_t} = \mathbf{K}_{a_t}\mathbf{x}_{a_t}$ to return updated $\mathbf{x}_{a_t}$
end for
return $\{a_t\}_{t=K+1}^{T}$

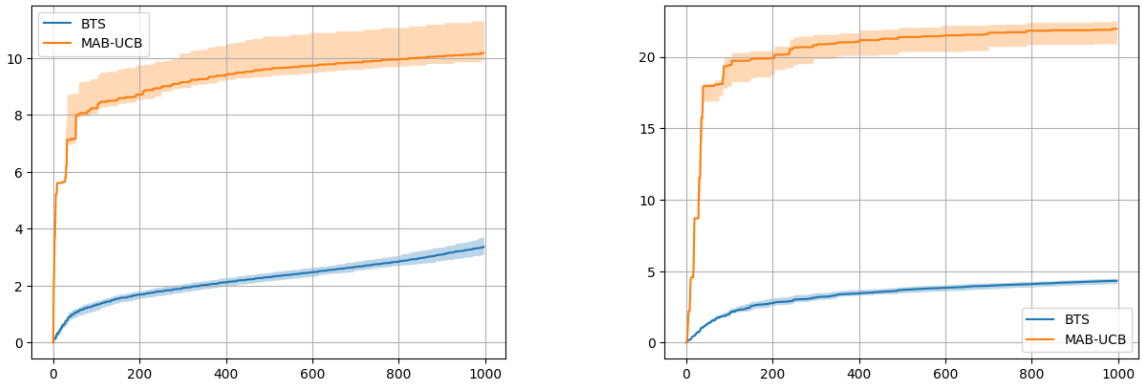
At this stage, we begin with sequentially sampling each of the arms for n consecutive time instants and capture the mean rewards. Using this finite length time series data for each arm we solve and form solutions to the convex optimization problem described in Equation (5.5). This gives us the estimates of period for each arm which we denote as $\hat{T}_k$.

Stage two: Regret minimization. In light of the periodic nature of arm rewards, we define the concept of a phase for arm k. We say that arm k is in phase p when the remainder of the epoch index t divided by its period $\hat{T}_k$ equals p, which is expressed as $p \equiv t(\mathrm{mod}\hat{T}_k)$. When an arm is in a particular phase, it exhibits the same mean reward, effectively categorizing it as an identical effective arm. In contrast, arms in different phases require separate mean reward learning processes, treating them as distinct effective arms. Therefore, we end up with $\hat{d} := \sum_k \hat{T}_k$ effective arms (unique mean rewards) to learn. Once, we know the number of effective arms, we employ the simple and effective upper confidence bound online learning framework to estimate the mean rewards.



(a) (b)

Figure 21. Regret $\mathcal{R}$ vs time t plots on two armed periodic bandits setting for (a) BTS-RaP and (b) MAB-UCB. Rewards of each arm is generated as per Equation (5.13).



(a) (b)

Figure 22. Regret $\mathcal{R}$ vs time t plots on two armed periodic bandits setting for MAB-UCB and BTS-RaP. Rewards of each arm is generated based on (5.12) with $\{p_1, p_2\}$ taking values (a) $\{7, 3\}$, (b) $\{9, 11\}$

5.5 Experimental Studies

We consider a two-armed bandit setup with three different experiments. In the first and second experiments, we represent the means of the two arms by:

$$\mu_1(t) = c + \sin\left(\frac{2\pi t}{p_1}\right), \ \mu_2(t) = c + \sin\left(\frac{2\pi t}{p_2}\right), \tag{5.12}$$

where, $t = \{1, 2, \ldots, T\}$ and c is some positive scalar. For the first experiment the tuple $\{p_1, p_2\}$ take the values {9, 11} and for the second one it takes the values {7, 3}.

For the third experiment, we consider periodic mixtures to generate rewards as follows,

$$\mu_1(t) = c + \sum_{i=1}^{3} \sin\left(\frac{2\pi t}{p_i}\right), \; \mu_2(t) = c + \sin\left(\frac{2\pi t}{p}\right), \tag{5.13}$$

where, for the first arm the periods are $\{p_1, p_2, p_3\} = \{3, 7, 11\}$ and second arm period is $p = 9$. In the first two experiments (Figures 21(a), (b)), we see that our proposed algorithm BTS-RaP outperforms MAB-UCB. While plotting the regret of MAB-UCB we do not consider the stage one (estimation of period) cost. One advantage of using RPT is that even if the period is large, we can still estimate it using RPT with fewer samples, sometimes, even when we have incomplete period length signal as illustrated in Figure 1. While MAB-UCB does achieve sub-linear regret it does so at a very slow pace compared to BTS-RaP. This is because we are selecting an optimal arm from a set of $\sum_k T_k$ arms and not 2 as stated in the problem. This increases the complexity of the MAB problem and is reflected in the regret curve.

The real issue is revealed in the third experiment where one of the arms rewards is a combination of sum of smaller periodic signals (7,3,11). Therefore, the resulting signal is a 231 length period signal. The second arm is a single period signal with period 9. So effectively, MAB-UCB algorithm has 240 effective arms to select from. Whereas, BTS-RaP effectively learns non-zero coordinates of the support vector $\mathbf{x}$ associated with each arm. This vector as highlighted earlier is sparse and the regret scales with the sum ℓ_0-norm of this support vector. Therefore, as seen in Figure 22, BTS-RaP achieves minimum regret quickly and MAB-UCB has to run for significantly longer time ($\sim 100\times$) to start learning the periodic pattern.

5.6 Conclusion

In this chapter, we consider bandits that exhibits periodicity. We incorporated the periodic structure of the rewards and proposed an algorithm to minimize the regret. To this end, we utilized the newly introduced, Ramanujan-based periodicity estimation techniques to sequentially update the estimate of the periods of each arm, and subsequently select the best arm at each time step. The results indicates that our RPT-based method dubbed BTS-RaP, can achieve sub-linear regret.

REFERENCES

Abari, Omid, Hariharan Rahul, and Dina Katabi. 2016. "Over-the-air function computation in sensor networks." *arXiv preprint arXiv:1612.02307.*

Abbasi-Yadkori, Y., D. Pal, and C. Szepesvari. 2012. "Online-to-confidence-set conversions and application to sparse stochastic bandits." In *Artificial Intelligence and Statistics,* 1–9.

Abbasi-Yadkori, Yasin, Dávid Pál, and Csaba Szepesvári. 2011. "Improved algorithms for linear stochastic bandits." *Advances in neural information processing systems* 24.

Abbasi-yadkori, Yasin, Dávid Pál, and Csaba Szepesvári. 2011. "Improved Algorithms for Linear Stochastic Bandits." In *Advances in Neural Information Processing Systems,* edited by J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, vol. 24. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2011/file/e1d5be1c7f2f456670de3d53c7b54f4a-Paper.pdf.

Aghazadeh, Amirali, Ryan Spring, Daniel LeJeune, Gautam Dasarathy, Anshumali Shrivastava, et al. 2018. "Mission: Ultra large-scale feature selection using count-sketches." In *International Conference on Machine Learning,* 80–88. PMLR.

Aghazadeh, Amirali, Ryan Spring, Daniel Lejeune, Gautam Dasarathy, Anshumali Shrivastava, and Richard Baraniuk. 2018. "MISSION: Ultra Large-Scale Feature Selection using Count-Sketches." In *Proceedings of the 35th International Conference on Machine Learning,* 80:80–88. October.

Ajalloeian, Ahmad, and Sebastian U. Stich. 2020a. "Analysis of SGD with Biased Gradient Estimators." *CoRR.*

———. 2020b. "Analysis of SGD with Biased Gradient Estimators." *CoRR* abs/2008.00051.

Aji, Alham Fikri, and Kenneth Heafield. 2017. "Sparse Communication for Distributed Gradient Descent." In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing,* 440–445.

Akhiezer, Naum Ilich, and Izrail Markovich Glazman. 2013. *Theory of linear operators in Hilbert space.* Courier Corporation.

Alistarh, Dan, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. 2017. "QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding." In *Advances in Neural Information Processing Systems,* vol. 30.

Amiri, Mohammad Mohammadi, and Deniz Gündüz. 2020. "Federated learning over wireless fading channels." *IEEE Transactions on Wireless Communications* 19 (5): 3546–3557.

Ang, Fan, Li Chen, Nan Zhao, Yunfei Chen, Weidong Wang, and F. Richard Yu. 2020. "Robust Federated Learning With Noisy Communication." *IEEE Transactions on Communications* 68 (6): 3452–3464.

Balzano, Laura, Robert Nowak, and Benjamin Recht. 2010. "High-dimensional robust subspace tracking under missing data and outliers." In *2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers,* 2087–2091. IEEE.

Baraniuk, R. G. 2007. "Compressive Sensing [Lecture Notes]." *IEEE Signal Processing Magazine* 24 (4): 118–121. https://doi.org/10.1109/MSP.2007.4286571.

Benedetto, Giuseppe Di, Vito Bellini, and Giovanni Zappella. 2020. *A Linear Bandit for Seasonal Environments.* arXiv: 2004.13576 [stat.ML].

Bernstein, Jeremy, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. 2018. "signSGD: Compressed Optimisation for Non-Convex Problems." In *Proceedings of the 35th International Conference on Machine Learning,* 80:560–569.

Besbes, Omar, Yonatan Gur, and Assaf Zeevi. 2019. "Optimal Exploration-Exploitation in a Multi-Armed-Bandit Problem with Non-Stationary Rewards." *Stochastic Systems* 9 (4): 319–337.

Besson, Lilian, and Emilie Kaufmann. 2019. "The generalized likelihood ratio test meets KLUCB: an improved algorithm for piece-wise non-stationary bandits." *Proceedings of Machine Learning Research vol XX* 1:35.

Bottou, Léon. 2010. "Large-scale machine learning with stochastic gradient descent." In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers,* 177–186. Springer.

Bottou, Léon, Frank E. Curtis, and Jorge Nocedal. 2018. "Optimization Methods for Large-Scale Machine Learning." *SIAM Review* 60 (2): 223–311.

Bubeck, Sébastien, Ofer Dekel, Tomer Koren, and Yuval Peres. 2015. "Bandit convex optimization:$\sqrt{T}$ regret in one dimension." In *Conference on Learning Theory,* 266–278. PMLR.

Cai, HanQin, Daniel McKenzie, Wotao Yin, and Zhenliang Zhang. 2022. "Zeroth-Order Regularized Optimization (ZORO): Approximately Sparse Gradients and Adaptive Sampling." *SIAM Journal on Optimization* 32 (2): 687–714.

Cai, Hengrui, Zhihao Cen, Ling Leng, and Rui Song. 2021. *Periodic-GP: Learning Periodic World with Gaussian Process Bandits.* arXiv: 2105.14422 [cs.LG].

Candès, Emmanuel J, Xiaodong Li, Yi Ma, and John Wright. 2011. "Robust principal component analysis?" *Journal of the ACM (JACM)* 58 (3): 1–37.

Candès, Emmanuel J, and Benjamin Recht. 2009. "Exact matrix completion via convex optimization." *Foundations of Computational mathematics* 9 (6): 717–772.

Carpentier, A., and R. Munos. 2012. "Bandit theory meets compressed sensing for high dimensional stochastic linear bandit." In *Artificial Intelligence and Statistics,* 190–198.

Charikar, Moses, Kevin Chen, and Martin Farach-Colton. 2002. "Finding Frequent Items in Data Streams." In *Automata, Languages and Programming,* 693–703. Berlin, Heidelberg.

Chen, Ningyuan, Chun Wang, and Longlin Wang. 2020. "Learning and Optimization with Seasonal Patterns." *CoRR* abs/2005.08088. arXiv: 2005.08088. https://arxiv.org/abs/2005.08088.

Chen, Qinyi, Negin Golrezaei, and Djallel Bouneffouf. 2023. *Dynamic Bandits with an Auto-Regressive Temporal Structure.* arXiv: 2210.16386 [cs.LG].

Chi, Yuejie, Yonina C Eldar, and Robert Calderbank. 2013. "PETRELS: Parallel subspace estimation and tracking by recursive least squares from partial observations." *IEEE Transactions on Signal Processing* 61 (23): 5947–5959.

Dani, Varsha, Thomas P Hayes, and Sham M Kakade. 2008. "Stochastic linear optimization under bandit feedback." *COLT* 2008:355–366.

Durand, Audrey, Charis Achilleos, Demetris Iacovides, Katerina Strati, Georgios D Mitsis, and Joelle Pineau. 2018. "Contextual bandits for adapting treatment in a mouse model of de novo carcinogenesis." *Proceedings of the 3rd machine learning for healthcare conference,* 67–82.

Garivier, Aurélien, and Eric Moulines. 2008. "On upper-confidence bound policies for non-stationary bandit problems." *arXiv preprint arXiv:0805.3415.*

Goetz, Jack, Kshitiz Malik, Duc Bui, Seungwhan Moon, Honglei Liu, and Anuj Kumar. 2019. "Active Federated Learning." *ArXiv.*

Goldenbaum, Mario, and Slawomir Stanczak. 2013. "Robust analog function computation via wireless multiple-access channels." *IEEE Transactions on Communications* 61 (9): 3863–3877.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning.* MIT press.

Gopalan, A., O.-A. Maillard, and M. Zaki. 2016. "Low-rank bandits with latent mixtures." *arXiv preprint arXiv:1609.01508.*

Gorbunov, Eduard, Dmitry Kovalev, Dmitry Makarenko, and Peter Richtárik. 2020. "Linearly converging error compensated SGD." *Advances in Neural Information Processing Systems* 33:20889–20900.

Graves, Alex. 2011. "Practical variational inference for neural networks." *Advances in neural information processing systems* 24.

Gupta, Neha, Ole-Christoffer Granmo, and Ashok Agrawala. 2011. "Thompson sampling for dynamic multi-armed bandits." In *2011 10th International Conference on Machine Learning and Applications and Workshops,* 1:484–489. IEEE.

Hu, Bin, Peter Seiler, and Laurent Lessard. 2021. "Analysis of biased stochastic gradient descent using sequential semidefinite programs." *Mathematical Programming* 187:383–408.

Ivkin, Nikita, Daniel Rothchild, Enayat Ullah, Vladimir braverman, Ion Stoica, and Raman Arora. 2019. "Communication-efficient Distributed SGD with Sketching." In *Advances in Neural Information Processing Systems,* vol. 32.

Jiang, Angela H., Daniel L. K. Wong, Giulio Zhou, David G. Andersen, Jeffrey Dean, Gregory R. Ganger, Gauri Joshi, et al. 2019. "Accelerating Deep Learning by Focusing on the Biggest Losers."

Jolliffe, Ian T, and Jorge Cadima. 2016. "Principal component analysis: a review and recent developments." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374 (2065): 20150202.

Juan, Yuchin, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. "Field-Aware Factorization Machines for CTR Prediction." In *Proceedings of the 10th ACM*

Conference on Recommender Systems, 43–50. Association for Computing Machinery.

Kairouz, Peter, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, et al. 2021. "Advances and Open Problems in Federated Learning." *Found. Trends Mach. Learn.* 14:1–210.

Karimireddy, Sai Praneeth, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. "SCAFFOLD: Stochastic Controlled Averaging for Federated Learning." In *Proceedings of the 37th International Conference on Machine Learning,* 119:5132–5143.

Karimireddy, Sai Praneeth, Quentin Rebjock, Sebastian U. Stich, and Martin Jaggi. 2019. "Error Feedback Fixes SignSGD and other Gradient Compression Schemes." *CoRR.*

Katharopoulos, A., and F. Fleuret. 2018. "Not All Samples Are Created Equal: Deep Learning with Importance Sampling." In *Proceedings of the International Conference on Machine Learning (ICML),* 80:2525–2534.

Khaled, Ahmed, Konstantin Mishchenko, and Peter Richtárik. 2019. *First Analysis of Local GD on Heterogeneous Data.*

Kleinberg, R., A. Slivkins, and E. Upfal. 2010. "Regret in the unknown stable-arm bandit problem." *arXiv preprint arXiv:1003.0722.*

Laguel, Yassine, Krishna Pillutla, Jérôme Malick, and Zaid Harchaoui. 2020. "Device Heterogeneity in Federated Learning: A Superquantile Approach." *ArXiv.*

Lale, Sahin, Kamyar Azizzadenesheli, Anima Anandkumar, and Babak Hassibi. 2019. "Stochastic linear bandits with hidden low rank structure." *arXiv preprint arXiv:1901.09490.*

Langford, John, and Tong Zhang. 2007. "The epoch-greedy algorithm for contextual multi-armed bandits." *Advances in neural information processing systems* 20 (1): 96–1.

Li, Lihong, Wei Chu, John Langford, and Robert E Schapire. 2010. "A contextual-bandit approach to personalized news article recommendation." In *Proceedings of the 19th international conference on World wide web,* 661–670.

Li, Qinbin, Yiqun Diao, Quan Chen, and Bingsheng He. 2021. "Federated Learning on Non-IID Data Silos: An Experimental Study." *CoRR* abs/2102.02079.

Li, Shuai, Alexandros Karatzoglou, and Claudio Gentile. 2016. "Collaborative filtering bandits." *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval,* 539–548.

Li, Tian, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. "Federated Optimization in Heterogeneous Networks." In *Proceedings of Machine Learning and Systems,* 2:429–450.

Li, Xiang, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2019. *On the Convergence of FedAvg on Non-IID Data.*

———. 2020. "On the Convergence of FedAvg on Non-IID Data." In *International Conference on Learning Representations (ICLR).*

Li, Xiang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2021. *Communication-Efficient Local Decentralized SGD Methods.* arXiv: 1910.09126 [stat.ML].

Liang, Yingyu, Zhuoran Xu, and Dale Schuurmans. 2019. "An exponential convergence rate for subspace estimation from partial observations." *arXiv preprint arXiv:1905.13595.*

Lopes, Miles E. 2016. "Unknown Sparsity in Compressed Sensing: Denoising and Inference." *IEEE Transactions on Information Theory* 62 (9): 5145–5166.

McMahan, H. Brendan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. 2016. "Federated Learning of Deep Networks using Model Averaging." *CoRR* abs/1602.05629.

Nishio, Takayuki, and Ryo Yonetani. 2019. "Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge." In *IEEE International Conference on Communications,* 1–7.

Oja, Erkki. 1982. "Simplified neuron model as a principal component analyzer." *Journal of mathematical biology* 15 (3): 267–273.

Qian, Xun, Peter Richtárik, and Tong Zhang. 2021. "Error compensated distributed SGD can be accelerated." *Advances in Neural Information Processing Systems* 34:30401–30413.

Ramanujan, S. 1918. "On certain trigonometrical sums and their applications in the theory of numbers." *Trans. Cambridge Philosoph. Soc* XXll (13): 259–276.

Re, Gerlando, Fabio Chiusano, Francesco Trovò, Diego Carrera, Giacomo Boracchi, and Marcello Restelli. 2021. "Exploiting history data for nonstationary multi-armed

bandit." In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part I 21,* 51–66. Springer.

Reddi, Sashank, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H. Brendan McMahan. 2021. *Adaptive Federated Optimization.* arXiv: 2003.00295 [cs.LG].

Ribero, Mónica, and Haris Vikalo. 2020. "Communication-efficient Federated Learning via Optimal Client Sampling." *ArXiv.*

Rothchild, Daniel, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora. 2020. "FetchSGD: Communication-Efficient Federated Learning with Sketching." In *Proceedings of the 37th International Conference on Machine Learning,* 119:8253–8265.

Ruan, Yichen, Xiaoxi Zhang, Shu-Che Liang, and Carlee Joe-Wong. 2020. "Towards Flexible Device Participation in Federated Learning for Non-IID Data." *ArXiv.*

Rusmevichientong, Paat, and John N Tsitsiklis. 2010. "Linearly parameterized bandits." *Mathematics of Operations Research* 35 (2): 395–411.

Saidi, P., A. Vosoughi, and G. K. Atia. 2019. "Detection of brain stimuli using Ramanujan periodicity transforms." *Journal of Neural Engineering* 16 (3): 036021.

Salehi, Farnood, Patrick Thiran, and Elisa Celis. 2018. "Coordinate Descent with Bandit Sampling."

Shah, Vatsal, Xiaoxia Wu, and Sujay Sanghavi. 2020. "Choosing the Sample with Lowest Loss Makes SGD Robust." In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS).*

Slivkins, Aleksandrs, et al. 2019. "Introduction to multi-armed bandits." *Foundations and Trends® in Machine Learning* 12 (1-2): 1–286.

Slivkins, Aleksandrs, and Eli Upfal. 2008. "Adapting to a Changing Environment: the Brownian Restless Bandits." In *COLT,* 343–354.

Stich, Sebastian U. 2019. "Unified optimal analysis of the (stochastic) gradient method." *arXiv preprint arXiv:1907.04232.*

Stich, Sebastian U, Jean-Baptiste Cordonnier, and Martin Jaggi. 2018. "Sparsified SGD with Memory." In *Advances in Neural Information Processing Systems,* vol. 31.

Tenneti, S. V., and P. P. Vaidyanathan. 2015. "Nested periodic matrices and dictionaries: New signal representations for period estimation." *IEEE Transactions on Signal Processing* 63 (14): 3736–3750.

———. 2016. "Detecting tandem repeats in DNA using Ramanujan filter bank." In *2016 IEEE International Symposium on Circuits and Systems (ISCAS),* 21–24. IEEE.

Tropp, J. A. 2004. "Greed is good: Algorithmic results for sparse approximation." *IEEE Transactions on Information Theory* 50 (10): 2231–2242.

Vaidyanathan, P. P. 2014. "Ramanujan sums in the context of signal processing—Part I : Fundamentals." *IEEE Transactions on Signal Processing* 62 (16): 4145–4157.

Villamediana, Jenely, Inés Küster, and Natalia Vila. 2019. "Destination engagement on Facebook: Time and seasonality." *Annals of Tourism Research* 79:102747.

Wang, Hongyi, Mikhail Yurochkin, Yuekai Sun, Dimitris S. Papailiopoulos, and Yasaman Khazaeni. 2020. "Federated Learning with Matched Averaging." *CoRR* abs/2002.06440.

Wang, Jianyu, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. 2020. "Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization." In *Advances in Neural Information Processing Systems,* 33:7611–7623.

Wang, Jianyu, Anit Kumar Sahu, Zhouyi Yang, Gauri Joshi, and Soummya Kar. 2019. "MATCHA: Speeding Up Decentralized SGD via Matching Decomposition Sampling." *CoRR* abs/1905.09435.

Wang, Yining, Yi Chen, Ethan X. Fang, Zhaoran Wang, and Runze Li. 2020. *Nearly Dimension-Independent Sparse Linear Bandit over Small Action Spaces via Best Subset Selection.* arXiv: 2009.02003 [stat.ML].

Wei, Xizixiang, and Cong Shen. 2021. "Federated Learning over Noisy Channels: Convergence Analysis and Design Examples." *CoRR.*

Wu, Jiaxiang, Weidong Huang, Junzhou Huang, and Tong Zhang. 2018. "Error Compensated Quantized SGD and its Applications to Large-scale Distributed Optimization." In *Proceedings of the 35th International Conference on Machine Learning,* 80:5325–5333. October.

Yang, Kai, Tao Jiang, Yuanming Shi, and Zhi Ding. 2020. "Federated learning via over-the-air computation." *IEEE Transactions on Wireless Communications* 19 (3): 2022–2035.

Yang, Tianbao, Raman Arora, and Jarvis Haupt. 2015. "Streaming principal component analysis." *Journal of Machine Learning Research* 16 (1): 2287–2320.

Yu, Felix X., Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. 2020. "Federated Learning with Only Positive Labels." *CoRR* abs/2004.10342.

Yu, Hsiang-Fu, Hung-Yi Lo, Hsun-Ping Hsieh, Jing-Kai Lou, Todd G McKenzie, Jung-Wei Chou, Po-Han Chung, Chia-Hua Ho, Chun-Fu Chang, Yin-Hsuan Wei, et al. 2010. "Feature engineering and classifier ensemble for KDD cup 2010." In *KDD cup.*

Yue, Yisong, Josef Broder, Robert Kleinberg, and Thorsten Joachims. 2012. "The k-armed dueling bandits problem." *Journal of Computer and System Sciences* 78 (5): 1538–1556.

Yurochkin, Mikhail, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. 2019. "Bayesian Nonparametric Federated Learning of Neural Networks." In *Proceedings of the 36th International Conference on Machine Learning,* 97:7252–7261.

Zhang, Junshan, Na Li, and Mehmet Dedeoglu. 2021. "Federated Learning over Wireless Networks: A Band-limited Coordinated Descent Approach." In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications,* 1–10.

Zhang, Junshan, Dong Zheng, and Mung Chiang. 2008. "The Impact of Stochastic Noisy Feedback on Distributed Network Utility Maximization." *IEEE Transactions on Information Theory* 54 (2): 645–665.

Zhou, Xiang, Yi Xiong, Ningyuan Chen, and Xuefeng Gao. 2021. "Regime switching bandits." *Advances in Neural Information Processing Systems* 34:4542–4554.

Zhu, Guangxu, Yong Wang, and Kaibin Huang. 2019. "Broadband analog aggregation for low-latency federated edge learning." *IEEE Transactions on Wireless Communications* 19 (1): 491–506.