© 2024 Lingda Wang

SEQUENTIAL PREDICTION AND DECISION MAKING, JOINT COMMUNITY DETECTION AND PHASE SYNCHRONIZATION, AND ACCELERATION METHODS FOR CONVEX OPTIMIZATION

BY

LINGDA WANG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2024

Urbana, Illinois

Doctoral Committee:

Associate Professor Zhizhen Jane Zhao, Chair
Associate Professor Ryan L. Sriver
Associate Professor Lav R. Varshney
Assistant Professor Ilan Shomorony

ABSTRACT

In this dissertation, several topics in machine learning are presented, including sequential prediction and decision making (e.g., time series/spatio-temporal sequences prediction and bandit learning), joint community detection and phase synchronization, and acceleration methods for convex optimization. Many applications such as demand/inventory planning, precipitation/climate prediction, online recommendation/advertising, web search, cryo-electron microscopy (cryo-EM) reconstruction, optimal transport, and video colocation have been shown to benefit from topics studied in this dissertation.

In Part I (Chapters 2, 3, 4, and 5), we focus on sequential prediction and decision making problems.

Chapters 2 and 3 focus on sequential prediction problems, including both time series and spatio-temporal sequences. Chapter 2 focuses on the problem of predicting sea surface temperature (SST) within the El Niño-Southern Oscillation (ENSO) region, which has been extensively studied recently due to its significant influence on global temperature and precipitation patterns. Statistical models such as linear inverse model (LIM), analog forecasting (AF), convolutional neural network (CNN), and recurrent neural network (RNN) have been widely used for ENSO prediction, offering flexibility and relatively low computational expense compared to large dynamic models. However, most of these models have limitations in capturing spatial patterns in SST variability or relying solely on linear dynamics. Chapter 2 presents a modified convolutional gated recurrent unit (ConvGRU) network for the ENSO region spatio-temporal sequence prediction problem, along with the Niño 3.4 index prediction as a down stream task. The proposed ConvGRU network, with an encoder-decoder sequence-to-sequence (Seq2Seq) structure, takes historical SST maps of the Pacific region as inputs and generates future SST maps for the subsequent months within the ENSO region. To evaluate the performance of the ConvGRU network, we train and test it using simulation and reanalysis datasets from multiple climate model ensembles, including a pre-industrial simulation spanning approximately 1300 years from the community climate system model version 4 (CCSM4) and a 30-member historical ensemble during 1921-2100 using the NOAA seamless system for prediction and earth system research (SPEAR) model. We also compare and contrast the prediction skill of the ConvGRU network against SOTA models. The results demonstrate that the ConvGRU network significantly improves the predictability of the Niño 3.4 index compared to existing statistical and deep learning prediction models, including LIM, AF, CNN, and RNN. This improvement is evidenced by extended useful prediction range, higher Pearson correlation (PC), lower root-mean-square error (RMSE), and lower weighted mean absolute percentage error (wMAPE). In Chapter 3, a practical and robust distribution forecast framework that relies on backtest-based bootstrap and adaptive residual selection is proposed. Distribution forecast can quantify forecast uncertainty and provide various forecast scenarios with their corresponding estimated probabilities. Accurate distribution forecast is crucial for demand planning when making production capacity or inventory allocation decisions. The proposed approach is robust to the choice of the underlying forecasting model, which accounts for uncertainty around the input covariates and relaxes the independence between residual and covariate assumptions. It reduces the absolute coverage error (ACE) by more than 63% compared to the classic bootstrap approaches and by $2\% - 32\%$ compared to a variety of state-of-the-art (SOTA) deep learning approaches on in-house product sales data and M4-hourly competition data.

Chapters 4 and 5 present two bandit learning problems. In Chapter 4, we consider a popular bandit model, cascading bandit (CB), for web search and online advertisement, where an agent aims to learn the K most attractive items out of a ground set of size L during the interaction with a user. Meanwhile, we take it a step further by considering CB in the piecewise-stationary environment where the user's preference may change over time. Two efficient algorithms, `GLRT-CascadeUCB` and `GLRT-CascadeKL-UCB`, are developed and shown to ensure regret upper bounds of $\mathcal{O}(\sqrt{NLT \log T})$, where N is the number of piecewise-stationary segments, and T is the length of time horizon. In addition, we show that the proposed algorithms are optimal (up to a logarithmic factor) by deriving a minimax lower bound of

$\Omega(\sqrt{NLT})$ for the piecewise-stationary CB. The efficiency of the proposed algorithms relative to existing approaches is validated through numerical experiments on both synthetic and real-world datasets. Chapter 5 studies the adversarial graphical contextual bandit problem, a variant of the adversarial multi-armed bandit problem, which leverages two categories of the most common side information: *contexts* and *side observations*. In this setting, an agent repeatedly chooses a set of L actions after being presented with a d-dimensional context vector. The agent not only incurs and observes the loss of the chosen action but also observes the losses of its neighboring actions in observation structures which are encoded as a series of feedback graphs. Two algorithms are developed based on `EXP3`. Under mild conditions, our analysis shows that for undirected feedback graphs the first algorithm, `EXP3-LGC-U`, achieves the regret of $\mathcal{O}(\sqrt{(L+\alpha(G)d)T\log L})$ where $\alpha(G)$ is the average *independence number* of the feedback graphs. A slightly weaker result is presented for the directed graph setting as well. The second algorithm, `EXP3-LGC-IX`, is developed for a special class of problems, for which the regret is reduced to $\mathcal{O}(\sqrt{\alpha(G)dT\log L\log(LT)})$ for both directed and undirected feedback graphs.

Part II (Chapter 6) studies the joint community detection and phase synchronization problem on the *stochastic block model with relative phase* where each node is associated with an unknown phase angle. This problem, with a variety of real-world applications, aims to recover the cluster structure and associated phase angles simultaneously. We show this problem exhibits a *multi-frequency* structure by closely examining its maximum likelihood estimation (MLE) formulation, whereas existing methods are not originated from this perspective. To this end, two simple yet efficient algorithms that leverage the MLE formulation and benefit from the information across multiple frequencies are proposed. The former is a spectral method based on the novel multi-frequency column-pivoted QR factorization. The factorization applied to the top eigenvectors of the observation matrix provides key information about the cluster structure and the associated phase angles. The second approach is an iterative multi-frequency generalized power method where each iteration updates the estimation in a matrix-multiplication-then-projection manner. Numerical experiments show that the proposed algorithms significantly improve the ability of exactly recovering the cluster structure and the accuracy of the estimated phase angles, compared to SOTA algorithms.

Part III of this dissertation (Chapters 7 and 8) focuses on acceleration methods for convex optimization problems. Chapter 7 introduces *almost tune-free* stochastic variance reduced gradient (SVRG) algorithm and stochastic recursive gradient (SARAH) algorithm equipped with i) Barzilai-Borwein (BB) step sizes; ii) averaging; and, iii) the inner loop length adjusted to the BB step sizes. In particular, SVRG, SARAH, and their BB variants are first re-examined through an *estimate sequence* lens to enable new averaging methods that tighten their convergence rates theoretically and improve their performance empirically when the step size or the inner loop length is chosen large. Then a simple yet effective means to adjust the number of iterations per inner loop is developed to enhance the merits of the proposed averaging schemes and BB step sizes. Chapter 8 introduces and analyzes a variant of the Frank Wolfe (FW) algorithm termed ExtraFW that has faster rate $\mathcal{O}(^1/_{k^2})$ on a class of machine learning problems where k is the iteration index. Compared with other parameter-free FW variants that have faster rates on the same problems, ExtraFW has improved rates and fine-grained analysis thanks to its prediction-correction update. Numerical experiments on binary classification with different sparsity-promoting constraints demonstrate that the empirical performance of ExtraFW is significantly better than FW and even faster than Nesterov's accelerated gradient on certain datasets. For matrix completion, ExtraFW enjoys smaller optimality gap and lower rank than FW.

To my friends, advisors, and parents, for their love and support.

ACKNOWLEDGMENTS

At the last moment of my Ph.D. student career, I would like to express my deepest and warmest gratitude to people who made my time at University of Illinois at Urbana-Champaign the most unforgettable journey of my life.

First and foremost, my deepest gratitude goes to my advisor, Prof. Zhizhen (Jane) Zhao. Jane is a brilliant and insightful researcher. She has been giving me not only solid research ideas but also the freedom to explore at my own. I learned a lot from her on how to find valuable research problems and solve them with proper methods. Jane is a perfect mentor. I cannot count how many times Jane offered dedicated guidance and helped to me when I encountered difficulties in research. Meanwhile, Jane has been caring about students beyond the research. This reminds me of not only being a good researcher but also a good person.

I would like to thank my doctoral and prelim committee members: Prof. Ryan Sriver, Prof. Lav R. Varshney, Prof. Ilan Shomorony, and Prof. Venugopal V. Veeravalli. Without their guidance and help, this dissertation would not have been possible. I would like to thank all other wonderful professors who offered me world class lectures including Prof. Matus Telgarsky, Prof. Olgica Milenkovic, Prof. Xiaochun Li, Prof. Negar Kiyavash, Prof. Alexander Schwing, Prof. Pierre Moulin, Prof. Yuguo Chen, and Prof. Niao He. Their lectures provided me with essential knowledge for research. I would also like to acknowledge Prof. Xiaolin Zhou, my undergraduate advisor at Fudan University, who encouraged and led me into my research career.

My great thanks also go to my research collaborators including Prof. Ryan Sriver, Prof. Vera Mikyoung Hur, Prof. Lav R. Varshney, Prof. Georgios B. Giannakis, Dr. Bingcong Li, Dr. Huozhi Zhou, and Savana Ammons, without whom works included in this dissertation would not have been possible. It is my great honor to talk, discuss, and work with you all.

I would like thank all the collegues, mentors, and managers during my internships at Amazon, especially Dr. Marshall Wang, Dr. Yan Lu, Abinaya Ulagappa, Dr. Haomin Yan, Jeremy Oldfather, Dr. Sitan Yang, Dr. Malcolm Wolff, Dr. Ruijun Ma, Michael Behrman, Dr. Dmitry Efimov, Dr. Kenny Shirley, Dr. Kari Torkkola, Dr. Abhishek Gupta, Dr. Keivan Majidi, and Prof. Matthew Plumlee. Your guidance and kindness, not only made my internships unforgettable but also helped me take the first steps toward industry.

I am very grateful to all my colleagues, labmates, and friends during these years, including but not limited to Dr. Yifeng Fan, Dr. Bingcong Li, Dr. Jinglin Chen, Dr. Huozhi Zhou, Yuqi Li, Dr. Junchi Yang, Anqi Liu, Xiaodan Hu, Yici Yan, Dr. Yifan Hu, Dr. Peijun Xiao, Dr. Siqi Zhang, Dr. Chao Pan, Dr. Chuanyi Zhang, Mengshen Yun, Chang Hu, Dr. Shuaichen Chang, Haoxian Chen, Jiaying Yang, Botao Chen, Weiji Qiu, Qianlong Wen, Hongyu Shen, Mengxiao Zhang, Binghui Wang, Dr. Liren Shan, Siyi Shen, Prof. Bihan Wen, Dr. Yanjun Li, and Prof. Yuheng Bu. Meanwhile, special thanks go to the office manager Peggy L. Wells for her help at Coordinated Science Lab.

Finally, I owe special gratitude to my parents, Mr. Lianbin Wang and Ms. Peizhu Li, for their endless love, encouragement, and support in my life. Their love and support will always provide me with confidence and power to embrace the future.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACE	Absolute Coverage Error
AF	Analog Forecasting
AFW	Accelerated Frank Wolfe
BA	Backtest-Additive
Bagging	Bootstrap Aggregating
BB	Barzilai-Borwein
BM	Backtest-Multiplicative
CB	Cascading Bandit
CCSM4	Community Climate System Model Version 4
CGS	Conditional Gradient Sliding
CM	Cascade Model
CMSW	Comparing Running Sample Means over A Sliding Window
CNN	Convolutional Neural Network
CO	Coverage
ConvGRU	Convolutional Gated Recurrent Unit
ConvLSTM	Convolutional LSTM
CPQR	Column-Pivoted QR
cryo-EM	Cryo-Electron Microscopy
CUSUM	Cumulative Sum
DF	Distribution Forecast

DFact	Deep Factors
DSSM	Deep State Space Models
ENSO	El Niño-Southern Oscillation
EPS	Error of Phase Synchronization
ES	Estimate Sequence
ERM	Empirical Risk Minimization
FC-LSTM	Fully-Connected LSTM
FC-RNN	Fully-Connected RNN
FFT	Fast Fourier Transform
FM	Fitted Models
FR	Fitted Residuals
FW	Frank Wolfe
FO	First-Order Oracle
GCB	Graphical Contextual Bandit
GD	Gradient Descent
GLRT	Generalized Likelihood Ratio Test
GPM	Generalized Power Method
IFFT	Inverse Fast Fourier Transform
IFO	Incremental First-Order Oracle
KAF	Kernel Analog Forecasting
KL	Kullback–Leibler
L-Avg	Last-Iteration Averaging
LIM	Linear Inverse Model
LMO	Linear Minimization Oracle
LR	Linear Regression
LSTM	Long Short-Term Memory
MAB	Multi-Armed Bandit

MAPE	Mean Absolute Percentage Error
MCAP	Minimum-Cost Assignment Problem
MF-CPQR	Multi-Frequency Column-Pivoted QR
MF-GPM	Multi-Frequency Generalized Power Method
MISO	Mixed-Integer Surrogate Optimization
ML	Machine Learning
MLE	Maximum Likelihood Estimation
MSE	Mean-Squared Error
NAG	Nesterov's Accelerated Gradient
NMME	North American Multi-Model Ensemble
NN	Neural Network
NP-hard	Non-Deterministic Polynomial-Time Hardness
PC	Prediction-Correction/Pearson Correlation
PF	Point Forecast
PHT	Page Hinkley Test
PMF	Probability Mass Function
QGB	Quantile Gradient Boosting
QLasso	Quantile Lasso
RF	Random Fores
RNN	Recurrent Neural Network
RMSE	Root-Mean-Square Error
SAG	Stochastic Average Gradient Algorithm
SARAH	Stochastic Recursive Gradient Algorithm
SBM	Stochastic Block Model
SBM-Ph	Stochastic Block Model with Relative Phase
SDCA	Stochastic Dual Coordinate Ascent Algorithm
SDP	Semidefinite Programming

Seq2Seq	Sequence-to-Sequence
SGD	Stochastic Gradient Descent
SOTA	State-of-the-Art
SPEAR	Seamless System for Prediction and Earth System Research
SRER	Success Rate of Exact Recovery
SST	Sea Surface Temperature
SVR	Support Vector Regression
SVRG	Stochastic Variance Reduced Gradient Algorithm
TFT	Temporal Fusion Transformer
U-Avg	Uniform Averaging
UCB	Upper Confidence Bound
W-Avg	Weighted Averaging
wMAPE	Weighted Mean Absolute Percentage Error

CHAPTER 1

INTRODUCTION

Machine learning (ML) is a field devoted to understanding and building methods that *learn*, that is, methods that leverage data to improve performance on some set of tasks [3]. It has fundamentally reshaped the world and improved most people's lives during the past decades, as tremendous progress and advancement have been made in this filed. Due to its wide applicability and remarkable performance, techniques developed by ML have been deeply embedded into real world applications, such as autonomous driving, recommendation systems, web search, speech recognition, and medicine design, to name a few.

As a cross-disciplinary field, ML covers plenty of topics across optimization, statistics, probability, modeling, etc. This dissertation, a summary of my Ph.D. research, presents several topics in ML, including sequential prediction and decision making, joint community detection and phase synchronization, and acceleration methods for convex optimization. In Part I , we study two sequential prediction problems and two sequential decision making problems, including robust nonparametric distribution forecast [4], spatio-temporal sequence model for climate prediction [5], cascading bandits [6] in the piecewise-stationary environment [7], and adversarial graphical contextual bandits [8]. In Part II, we study on an emerging problem, joint community detection and phase (group) synchronization [9], which aims to recover the cluster structure and the associated phase (group) simultaneously. For convex optimization (Part III), we mainly focus on the acceleration methods, in which we introduces *almost tune-free* stochastic variance reduction algorithms [10] and an accelerated Frank-Wolfe (FW) algorithm [11].

1.1 Sequential Prediction and Decision Making

1.1.1 Convolutional GRU Network for Seasonal Prediction of the El Niño-Southern Oscillation

The El Niño-Southern Oscillation (ENSO) phenomenon over the tropical Pacific region is the most energetic driver of climate variability on the seasonal to interannual timescales [12]. It can significantly influence global oceanic and atmospheric dynamics, particularly during its irregular warming (El Niño) and cooling (La Niña) phases. The impacts of ENSO are widespread, leading to anomalous temperature and precipitation patterns on a global scale [13, 14, 15], as well as causing extreme and hazardous weather conditions on regional scales, such as winter to early spring tornado outbreaks in the United States [16], tropical cyclone intensity changes in northwestern Pacific [17], and unusual fire weather in Australia [18] and the United States [19]. Consequently, accurate prediction of sea surface temperature (SST) maps within the ENSO region and its associated Niño indices—for instance, Niño 1+2, 3, 3.4, and 4 [20, 21, 22]—has become a critical area of research. Reliable ENSO prediction can provide valuable insights for decision-making processes in various sectors, including government agencies, food and insurance industries, and transportation, enabling them to prepare for the associated impacts [23, 24].

Prediction models for SST maps within the ENSO region and the associated Niño indices can be broadly classified into two types: dynamical models and statistical models. Dynamical models, such as the north American multi-model ensemble (NMME) [25], are commonly used for the seasonal ENSO prediction. However, these model ensembles are computationally intensive, sensitive to initialization conditions, and thus expensive to run. Consequently, dynamical models usually require multiple model runs with various initialization conditions with the help of supercomputers. In contrast, Chapter 2 focuses on the statistical models due to their simplicity and comparable prediction skill to dynamical models [26, 27, 28]. One widely used statistical ENSO prediction model is the linear inverse model (LIM) [29, 30], which employs principal components analysis and Markov prediction to approximate trends and predict future states based on empirical orthogonal functions, similar to linear regression (LR) [28]. However, LIM fails to capture nonlinear

ENSO dynamics—for instance, surface-subsurface interactions and surface winds [31]—and can lead to underestimation of ENSO variability. Another type of statistical prediction model is based on Lorenz's analog forecasting (AF) [32]. Initially, AF based models use observed or free-running model data as libraries of states. Predictions are then generated by matching states in the library that are very similar to observed data at prediction initialization and follow the evolution on these so-called analogs. Advantages of AF based models include avoiding expensive and unstable initialization systems and reducing structural model error. The recently introduced kernel analog forecasting (KAF) model [33, 28, 34], as a generalization of conventional AF based models, utilizes nonlinear kernels to better capture nonlinearity in ENSO dynamics. Recently, with the development of deep learning techniques, the convolutional neural network (CNN) [35] and long short-term memory (LSTM) network [36, 37] have been used for predicting Niño indices, but their prediction skills have hardly been extended to predict spatial patterns in SST variability within the ENSO region.

In Chapter 2, we propose the use of a convolutional gated recurrent unit (ConvGRU) network, inspired by and modified from the original developments [38, 39, 40], to predict SST maps within the ENSO region, along with the Niño 3.4 index as a downstream task, which is the most commonly used index to define El Niño and La Niña events [22]. The ConvGRU network has an encoder-decoder sequence-to-sequence (Seq2Seq) structure, with both the encoder and the decoder consisting of multi-layer ConvGRU cells. The encoder compresses the input SST maps of the Pacific region into hidden states across all layers, and the decoder unfolds the hidden states from the encoder to generate predictions within the ENSO region. The ConvGRU cell, a key component of both the encoder and the decoder, incorporates several 2D convolutional layers. This architecture enables the ConvGRU network to take historical SST maps of the Pacific region as inputs and generate future SST maps of the ENSO region for subsequent months, taking into consideration of spatio-temporal correlation of the SST maps. Moreover, this architecture significantly reduces the number of network parameters while accelerating the training process.

To evaluate the performance of the ConvGRU network, we conduct numerical experiments and compare it against existing models, such as KAF, LIM, Seq2Seq with GRU, LR, and CNN using global climate ensembles and atmospheric reanalysis datasets. These datasets include two SST simulation datasets and one surface air temperature reanalysis dataset. The comparison results demonstrate that the ConvGRU network achieves significant improvements over other models in terms of useful prediction range, Pearson correlation (PC), root-mean-square error (RMSE), and weighted mean absolute percentage error (wMAPE).

By developing an improved prediction model that accurately captures the complex dynamics and spatial patterns of SST within the ENSO region, Chapter 2 aims to contribute to better understanding and prediction of ENSO-related climate phenomena. Further research can explore further enhancements to the network architecture and investigate its applicability to other climate-related features and prediction tasks.

1.1.2 Robust Nonparametric Distribution Forecast with Backtest-based Bootstrap and Adaptive Residual Selection

Time series forecasting is crucial in many industrial applications and enables data-driven planning [41, 42, 43], such as making production capacity or inventory allocation decisions based on demand forecast [44]. Planners or optimization systems that consume the forecast often require the estimated distribution of the response variable (referred to as the distribution forecast, or the DF) instead of only the estimated mean/median (referred to as the point forecast, or the PF) to make informed and nuanced decisions. An accurate DF method should ideally factor in different sources of forecast uncertainty, including uncertainty associated with parameter estimates and model misspecification [42]. Furthermore, when deploying a DF model in industrial applications, there are other important practical considerations such as the ease of adoption, latency, interpretability, and robustness to model misspecification. To this end, a practical and robust DF framework that uses backtesting [45] is proposed in Chapter 3 to build a collection of predictive residuals and an adaptive residual selector to pick the relevant residuals for bootstrapping DF.

We empirically evaluate the performance of various DF approaches on our in-house product sales data and the M4-hourly [46, 47] competition data. The proposed DF approach reduces the absolute coverage error (ACE) by more than 63% compared to the classic bootstrap approaches and by $2\% - 32\%$ compared to a variety of state-of-the-art (SOTA) deep learning approaches.

1.1.3 Piecewise-Stationary Cascading Bandits

Online recommendation [48] and web search [49, 50] are of significant importance for the modern economy. Based on a user's browsing history, these systems strive to maximize satisfaction and minimize regret by presenting the user with a list of items (e.g., web pages and advertisements) that meet her/his preference. Such a scenario can be modeled via cascading bandits (CB) [6], where an agent aims to identify the K most attractive items out of total L items contained in the ground set. The learning task proceeds sequentially, where per time slot, the agent recommends a ranked list of K items and receives the reward and feedback on which item is clicked by the user.

CB can be viewed as multi-armed bandits (MAB) tailored for the cascade model (CM) [51], where CM models a user's online behavior. Existing works on CB [6, 52] can be categorized according to whether stationary or non-stationary environment is studied. In stationary environments, the attraction distributions of items do not evolve over time. On the other hand, non-stationary environments are prevalent in real-world applications such as web search, online advertisement, and recommendation since user's preference is time-varying [53, 54, 55]. Algorithms designed for stationary scenarios can suffer from a linear regret when applied to non-stationary environments directly [56, 57].

Chapter 4 focuses on the piecewise-stationary environment, where the user's preference remains stationary over some number of time slots, named *piecewise-stationary segments*, but can shift abruptly at some unknown times, called *change-points*. To address the piecewise-stationary environment, one can either choose *passively adaptive approaches* [57, 58, 59] or *actively adaptive approaches* [60, 61, 62, 63]. Passively adaptive approaches ignore when a change-point occurs. For active adaptive approaches, a change-point detection algorithm such as CUSUM [64, 61], Page Hinkley Test (PHT) [65, 61], or comparing running sample means over a sliding window (CMSW) [60] is included. Within the area of piecewise-stationary CB, only passively adaptive approaches have been studied [56]. In Chapter 4, we introduce the generalized likelihood ratio test (GLRT) [66, 62] for actively adaptive CB algorithms. In particular, we develop two GLRT based algorithms `GLRT-Cascade-UCB` and `GLRT-CascadeKL-UCB` to enhance both theoretical and practical effectiveness for piecewise-stationary CB.

1.1.4 Adversarial Graphical Contextual Bandits

Since the classical MAB does not fully leverage the widely available side information, it is not delicate enough for real world applications. This has motivated studies on *contextual bandits* [67, 68, 69, 70] and *graphical bandits* [71, 72, 73, 74], which aim to address two categories of the most common side information, *contexts* and *side observations*, respectively. In a contextual bandit problem, a learning agent chooses an action to play based on the context for the current time slot and the past interactions. In a graphical bandit setup, playing an action not only discloses its own loss, but also the losses of its neighboring actions. Applications of contextual bandits include mobile health [75] and online personalized recommendation [67, 76, 77], whereas applications of graphical bandits include viral marketing, online pricing, and online recommendation in social networks [74, 78].

However, contextual or graphical bandits alone may still not capture many aspects of real-world applications in social networks efficiently. As a motivating example, consider the viral marketing over a social network where a salesperson aims to investigate the popularity of a series of products [79]. At each time slot, the salesperson could offer a survey (context) of some product to a user together with a promotion. The salesperson also has a chance to survey the user's followers (side observations) in this social network which can be realized by assuming that i) if the user would like to get the promotion, the user should finish the questionnaire and share it in the social network, and ii) if the followers would like to get the same promotion, they need to finish the same questionnaire shared by the user.

Chapter 5 presents the first study on adversarial linear contextual bandits with graph-structured side observations (or simply, graphical contextual bandits). Specifically, at each time slot t, an adversary chooses the loss vector for each action in a finite set of L actions, and then a learning agent chooses from this L-action set after being presented with a d-dimensional context. After playing the chosen action, the agent not only incurs and observes the loss of the chosen action, but also observes losses of its neighboring action in the feedback graph G_t, where the losses are generated by the contexts and loss vectors under the linear payoff assumption [80]. The goal of the agent is to minimize the regret, defined as the gap between the losses incurred by the agent and that of some suitable benchmark policy. Under mild conditions, we develop two algorithms for this problem with theoretical guarantees: i) `EXP3-LGC-U`, inspired by `EXP3-SET` [72, 74] and `LinEXP3` [81]; ii) `EXP3-LGC-IX`, inspired by `EXP3-IX` [82] and `LinEXP3`.

1.2 Multi-Frequency Joint Community Detection and Phase Synchronization

Community detection on *stochastic block model* (SBM) [83] and *phase synchronization* [84], are both of fundamental importance among multiple fields, such as ML [85, 86], social science [87, 88], and signal processing [89, 90, 91], to name a few.

Community detection on SBM. Consider the symmetric SBM with N nodes that fall into M underlying clusters of equal size $s = {}^{N}\!/_{M}$. SBM generates a random graph $\mathcal{G}$ such that each pair of nodes (i, j) are connected independently with probability p if (i, j) belong to the same cluster and with probability q otherwise. The goal is to recover underlying cluster structure of nodes, given the adjacency matrix $\boldsymbol{A}_{\mathrm{SBM}} \in \{0, 1\}^{N\times N}$ of the observed graph $\mathcal{G}$. During the past decade, significant progress has been made on the information-theoretic threshold of the exact recovery on SBM [92, 93, 83], in the regime where $p = {}^{\alpha \log N}\!/_{N}$, $q = {}^{\beta \log N}\!/_{N}$, and $\sqrt{\alpha} - \sqrt{\beta} > \sqrt{M}$. The maximum likelihood estimation (MLE) formulation of community detection on SBM

$$\max_{\boldsymbol{H}\in\mathcal{H}} \quad \left\langle \boldsymbol{A}_{\mathrm{SBM}}, \boldsymbol{H}\boldsymbol{H}^\top \right\rangle, \tag{1.1}$$

is capable of achieving the exact recovery in the above regime, where $\mathcal{H} := \{\boldsymbol{H} \in \{0, 1\}^{N\times M} : \boldsymbol{H}\mathbf{1}_M = \mathbf{1}_N, \boldsymbol{H}^\top\mathbf{1}_N = s\mathbf{1}_M\}$ is the feasible set. However, the MLE (1.1) is non-convex and NP-hard in the worst case. Therefore, different approaches based on MLE (1.1) or other formulations are proposed to tackle this problem such as spectral method [94, 95, 96, 97, 98, 99, 100, 101], semidefinite programming (SDP) [92, 102, 103, 104, 105, 106, 107, 108, 109], and belief propagation [93, 110].

Phase synchronization. The phase synchronization problem concerns recovering phase angles $\theta_1, \ldots, \theta_N$ in $[0, 2\pi)$ from a subset of possibly noisy phase transitions $\theta_{ij} := (\theta_i - \theta_j) \mod 2\pi$. The phase synchronization problem can be encoded into an observation graph $\mathcal{G}$ where each phase angle is associated with a node i, and the phase transitions are observed between θ_i and θ_j if and only if there is an edge in $\mathcal{G}$ connecting the pair of nodes (i, j). Under the random corruption model [84, 111], observations constitute a Hermitian matrix whose (i, j)th entry for any $i < j$ satisfies,

$$\boldsymbol{A}_{\mathrm{Ph},ij} = \begin{cases} e^{\iota(\theta_i - \theta_j)}, & \text{with probability } r \in [0, 1), \\ u_{ij} \sim \mathrm{Unif}(U(1)), & \text{with probability } 1 - r, \end{cases}$$

where $\iota = \sqrt{-1}$ is the imaginary unit, and $U(1)$ is unitary group of dimension 1. The most common formulation of the phase synchronization problem is through the following nonconvex optimization program

$$\max_{\boldsymbol{x} \in \mathbb{C}_1^N} \quad \left\langle \boldsymbol{A}_{\mathrm{Ph}}, \boldsymbol{x}\boldsymbol{x}^{\mathsf{H}} \right\rangle, \tag{1.2}$$

where $\mathbb{C}_1^N$ is the Cartesian product of N copies of $U(1)$. Again, similar to SBM, solving (1.2) is non-convex and NP-hard [112]. Many algorithms have been proposed for practical and approximate solutions of (1.2), including spectral and SDP relaxations [84, 113, 114, 115, 116], and generalized power method (GPM) [117, 118, 119]. Besides, [120, 121, 122] consider the phase synchronization problem in multiple frequency channels which in general outperforms the formulation (1.2).

Recently, an increasing interest [123, 1, 2] has been seen in the *joint community detection and phase (or group) synchronization problem* (joint estimation problem, for brevity). As illustrated in Figure 1.1, the joint estimation problem assumes data points associated with phase angles (or group elements) in a network fall into M underlying clusters, and aims to simultaneously recover the cluster structure and the associated phase angles (or group elements). The joint estimation problem is motivated by the 2D class

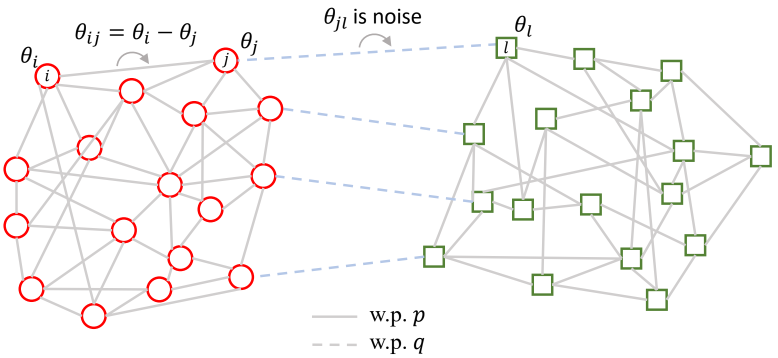


Figure 1.1: Illustration of the joint estimation problem on a network with two clusters of equal size. Each node is associated with a phase angle. Each pair of nodes within the same cluster (resp. across clusters) are independently connected with probability p (resp. q) as shown in solid (resp. dash) lines. Also, a phase transition $\theta_{ij} = \theta_i - \theta_j$ (resp. θ_{ij} is noise) is observed on each edge (i, j) within each cluster (resp. across clusters). The goal is to recover the cluster structure and the associated phase angles simultaneously.

averaging procedure in cryo-electron microscopy (cryo-EM) single particle reconstruction [124, 89, 90], which aims to cluster 2D projection images taken from similar viewing directions, align ($U(1)$ or $SO(2)$ synchronization due to the in-plane rotation) and average projection images in each cluster to improve their signal-to-noise ratio.

Chapter 6 studies the joint estimation problem based on the probabilistic model, *stochastic block model with relative phase* (SBM-Ph), which is similar to the probabilistic model considered in recent publications [1, 2, 123]. Specifically, given N nodes in a network assigned into M underlying clusters of equal size $s = {}^{N}/_{M}$, we assume that each node i is associated with an unknown phase angle $\theta_i^* \in \Omega$, where Ω is a discretization of $[0, 2\pi)$[1]. For each pair of nodes (i, j), if they belong to the same cluster, their phase transition $\theta_{ij} := (\theta_i - \theta_j) \mod 2\pi$ can be obtained with probability p; otherwise, we obtain noise generated uniformly at random from Ω with probability q. The goal of the joint estimation problem is to simultaneously recover the cluster structure and the associated phase angles. This problem can be formulated as an optimization program maximizing not only the edge connections inside each cluster, but also the consistency among the observed phase transitions within

[1] The joint estimation problem is also extended into $[0, 2\pi)$ in Section 6.2.3.

each cluster. Still, such kind of optimization programs, similar to community detection on SBM (1.1) and phase synchronization (1.2), is non-convex. In [123, Fan et al., 2022], an SDP based method is proposed to achieve approximate solutions with a polynomial computational complexity. [1, Fan et al., 2023] proposes a spectral method based on the block-wise column-pivoted QR (CPQR) factorization which scales linearly with the number of edges in the network. The most recent work [2] develops an iterative GPM where each iteration follows a matrix-multiplication-then-projection manner. The iterative GPM requires an initialization and the computational complexity of each iteration also scales linearly with the number of edges in the network. However, existing methods are not developed from the MLE perspective which limits their performance on the joint estimation problem.

Unlike existing methods, Chapter 6 studies the joint estimation problem by first closely examining its MLE formulation which exhibits a *multi-frequency* structure (detailed in Section 6.2). More specifically, the MLE formulation is maximizing the summation over multiple frequency components whose first frequency component is actually the objective function studied in [123, Fan et al., 2022], [1, Fan et al., 2023], and [2, Chen et al, 2021]. Based on the new insight, *a spectral method based on the multi-frequency column-pivoted QR* (MF-CPQR) *factorization* and *an iterative multi-frequency generalized power method* (MF-GPM) are proposed to tackle the MLE formulation of the joint estimation problem, and both significantly outperform SOTA methods in numerical experiments.

1.3 Acceleration Methods for Convex Optimization

1.3.1 Almost Tune-Free Variance Reduction

Consider the empirical risk minimization (ERM) problem,

$$\min_{\mathbf{x}\in\mathbb{R}^d} \quad f(\mathbf{x}) := \frac{1}{n}\sum_{i\in[n]} f_i(\mathbf{x}), \tag{1.3}$$

where $\mathbf{x} \in \mathbb{R}^d$ is the parameter vector to be learned from data; the set $[n] := \{1, 2, \ldots, n\}$ collects data indices; and, f_i is the loss function associated with datum i. Suppose that f is μ-strongly convex and has L-Lipchitz continuous gradient. The condition number of f is denoted by $\kappa := L/\mu$. Throughout, $\mathbf{x}^*$ denotes the optimal solution of (1.3). The standard approach to solve (1.3) is *gradient descent* (GD) [125], which updates the decision variable via

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \nabla f(\mathbf{x}_k),$$

where k is the iteration index and η the step size (or learning rate). For a strongly convex f, GD convergences linearly to $\mathbf{x}^*$, that is, $\|\mathbf{x}_k - \mathbf{x}^*\|^2 \leq (c_\kappa)^k \|\mathbf{x}_0 - \mathbf{x}^*\|^2$ for some κ-dependent constant $c_\kappa \in (0, 1)$ [125].

In the big data regime however, where n is large, obtaining the gradient per iteration can be computationally prohibitive. To cope with this, the *stochastic gradient descent* (SGD) reduces the computational burden by drawing uniformly at random an index $i_k \in [n]$ per iteration k and adopting $\nabla f_{i_k}(\mathbf{x}_k)$ as an estimate of $\nabla f(\mathbf{x}_k)$. Albeit computationally lightweight with the simple update

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \nabla f_{i_k}(\mathbf{x}_k),$$

the price paid is that SGD comes with sublinear convergence, hence slower than GD [126, 127]. It has been long recognized that the variance $\mathbb{E}[\|\nabla f_{i_t}(\mathbf{x}_t) - \nabla f(\mathbf{x}_t)\|^2]$ of the gradient estimate affects critically SGD's convergence slowdown.

This naturally motivated gradient estimates with *reduced variance* compared with SGD's simple $\nabla f_{i_k}(\mathbf{x}_k)$. A gradient estimate with reduced variance can be obtained by capitalizing on the finite sum structure of (1.3). One idea is to judiciously evaluate a so-termed *snapshot gradient* $\nabla f(\mathbf{x}_s)$ and use it as an anchor of the stochastic draws in subsequent iterations. Members of the variance reduction family include stochastic variance reduced gradient algorithm (SVRG) [128], stochastic average gradient algorithm (SAG) [129], SAGA [130], mixed-integer surrogate optimization (MISO) [131], stochastic recursive gradient algorithm (SARAH) [132], and their variants [133, 134, 135, 136]. Most of these algorithms rely on the update $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \mathbf{v}_k$, where η is a constant step size and $\mathbf{v}_k$ is an algorithm-specific gradient estimate that takes advantage of the snapshot gradient. In Chapter 7, SVRG and SARAH are of central interest because they are memory efficient compared with SAGA and have no requirement for the duality arguments that stochastic dual coordinate ascent algorithm (SDCA) [137] entails. Variance reduction methods converge linearly when f is strongly convex. To fairly compare the complexity of (S)GD with that of variance reduction algorithms which combine snapshot gradients with the stochastic ones, we will rely on the incremental first-order oracle (IFO) [138].

Definition 1.1. *An IFO takes f_i and $\mathbf{x} \in \mathbb{R}^d$ as input and returns the (incremental) gradient $\nabla f_i(\mathbf{x})$.*

For convenience, IFO complexity is abbreviated as complexity in Chapter 7. A desirable algorithm obtains an ϵ-accurate solution satisfying $\mathbb{E}[\|\nabla f(\mathbf{x})\|^2] \leq \epsilon$ or $\mathbb{E}[f(\mathbf{x}) - f(\mathbf{x}^*)] \leq \epsilon$ with minimal complexity for a prescribed ϵ. Complexity for variance reduction alternatives such as SVRG and SARAH is $\mathcal{O}\big((n+\kappa)\ln\frac{1}{\epsilon}\big)$, a clear improvement over GD's complexity $\mathcal{O}\big(n\kappa\ln\frac{1}{\epsilon}\big)$. When high accuracy (small ϵ) is desired, the complexity of variance reduction algorithms is also lower than SGD's complexity of $\mathcal{O}\big(\frac{1}{\epsilon}\big)$. Though theoretically appealing, SVRG and SARAH entail grid search to tune the step size which is often painstakingly hard and time consuming. An automatically tuned step size for SVRG was introduced by Barzilai-Borwein (BB) [139, 140]. However, since both SVRG and SARAH have a double-loop structure, the inner loop length also requires tuning in addition to the step size. Other works relying on BB step sizes introduce additional tunable parameters on top of the inner loop length [141]. In a nutshell, *tune-free* variance reduction algorithms still have desired aspects to investigate and fulfill.

Along with the BB step sizes, Chapter 7 establishes that in order to obtain *tune-free* SVRG and SARAH schemes, one must: i) develop novel types of averaging, and ii) adjust the inner loop length along with step size as well. Averaging in double-loop algorithms reflects the means of choosing the starting point of the next outer loop [128, 140, 132]. The types of averaging considered so far have been employed as tricks to simplify proofs while in the algorithm itself, the last iteration is the most prevalent choice for the starting point of the ensuing outer loop. However, we contend that different averaging methods result in different performance. The best averaging depends on the choice of other parameters. In addition to averaging, we argue that the choice of the inner loop length for BB-SVRG in [140] is too pessimistic. Addressing this with a simple modification leads to the desired *almost tune-free* SVRG and SARAH.

1.3.2 Parameter-Free Frank Wolfe with Faster Rates

Consider the optimization problem

$$\min_{\mathbf{x}\in\mathcal{X}} \quad f(\mathbf{x}), \tag{1.4}$$

where f is a smooth convex function, while the constraint set $\mathcal{X} \subset \mathbb{R}^d$ is assumed to be convex and compact, and d is the dimension of the variable $\mathbf{x}$. Throughout we denote by $\mathbf{x}^* \in \mathcal{X}$ a minimizer of (1.4). For many ML and signal processing problems, the constraint set $\mathcal{X}$ can be structural but it is difficult or expensive to project onto. Examples include matrix completion in recommendation systems [142] and image reconstruction [143], whose constraint sets are nuclear norm ball and total-variation norm ball, respectively. The applicability of projected GD [125] and Nesterov's accelerated gradient (NAG) [144, 145, 146] is thus limited by the computational barriers of projection, especially as d grows large.

An alternative to GD for solving (1.4) is the Frank Wolfe (FW) algorithm [147, 148, 149], also known as the *conditional gradient* method. FW circumvents the projection in GD by solving a subproblem with a *linear* loss per iteration. For a structural $\mathcal{X}$, such as the constraint sets mentioned earlier, it is possible to solve the subproblem either in closed form or through low-complexity numerical methods [148, 150], which saves computational cost relative to projection. In addition to matrix completion and image reconstruction, FW has been appreciated in several applications including structural support vector machine [151], video colocation [152], optimal transport [153], and submodular optimization [154], to name a few.

Although FW has well documented merits, it exhibits slower convergence when compared to NAG. Specifically, FW satisfies $f(\mathbf{x}_k) - f(\mathbf{x}^*) = \mathcal{O}(\frac{1}{k})$, where the subscript k is iteration index. This convergence slowdown is confirmed by the lower bound which indicates that the number of FW subproblems to solve in order to ensure $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \epsilon$ is no less than $\mathcal{O}\big(\frac{1}{\epsilon}\big)$ [155, 148]. Thus, FW is a lower-bound-matching algorithm in general. However, improved FW type algorithms are possible either in empirical performance or in speedup rates for certain subclasses of problems. Next, we deal with these improved rates paying attention to whether implementation requires knowing parameters such as the smoothness constant or the diameter of $\mathcal{X}$.

Parameter-dependent FW with faster rates. This class of algorithms utilizes parameters that are obtained for different instances of f and $\mathcal{X}$. Depending on the needed parameters, these algorithms are further classified into: i) line search based FW, ii) shorter step size aided FW, and iii) conditional gradient sliding (CGS). Line search based FW relies on $f(\mathbf{x})$ evaluations which renders inefficiency when acquisition of function values is costly. The vanilla FW with line search converges with rate $\mathcal{O}(\frac{1}{k})$ on general problems [148]. Jointly leveraging line search and *away steps*, variants of FW converge linearly for strongly convex problems when $\mathcal{X}$ is a polytope [156, 149]; see also [157, 158]. To improve the memory efficiency of away steps, a variant is further developed in [159, Garber et al., 2016]. Shorter step sizes refer to those used in [160, Levitin et al, 1966] and [150, Garber et al., 2015], where the step size is obtained by minimizing an 1D quadratic function over $[0, 1]$. Shorter step sizes require the smoothness parameter, which needs to be estimated for different loss functions. If $\mathcal{X}$ is strongly convex and the optimal solution is at the boundary of $\mathcal{X}$, it is known that FW converges linearly [160]. For uniformly (and thus strongly) convex sets, faster rates are attained given that the optimal solution is at the boundary of $\mathcal{X}$ [161]. When both f and $\mathcal{X}$ are strongly convex, FW with shorter step size converges at a rate of $\mathcal{O}(\frac{1}{k^2})$, regardless of where the optimal solution resides [150]. The last category is CGS, where both smoothness parameter and the diameter of $\mathcal{X}$ are necessary. In CGS, the subproblem of the original NAG that relies on projection is replaced by gradient sliding that solves a sequence of FW subproblems. A faster rate $\mathcal{O}(\frac{1}{k^2})$ is obtained at the price of: i) requiring at most $\mathcal{O}(k)$ FW subproblems in the kth iteration and ii) an inefficient implementation since the NAG subproblem has to be solved up to a certain accuracy.

Parameter-free FW. The advantage of a parameter-free algorithm is its efficient implementation. Since no parameter is involved, there is no concern on the quality of parameter estimation. This also saves time and effort because the step sizes do not need tuning. Although implementation efficiency is ensured, theoretical guarantees are challenging to obtain. This is because

$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k)$ cannot be guaranteed without line search or shorter step sizes. Faster rates for parameter-free FW are rather limited in number, and most of existing parameter-free FW approaches rely on diminishing step sizes at the order of $\mathcal{O}(\frac{1}{k})$. For example, the behavior of FW when k is large and $\mathcal{X}$ is a polytope is investigated under strong assumptions on $f(\mathbf{x})$ to be twice differentiable and locally strongly convex around $\mathbf{x}^*$ [162]. Accelerated FW (AFW) [163] replaces the subproblem of NAG by a single FW subproblem, where constraint-specific faster rates are developed. Taking an active ℓ_2 norm ball constraint as an example, AFW guarantees a rate of $\mathcal{O}\big(\frac{\ln k}{k^2}\big)$. A natural question is whether the $\ln k$ in the numerator can be eliminated. In addition, although the implementation involves no parameter, the analysis of AFW relies on the value $\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$.

Aiming at parameter-free FW with faster rates (on certain constraints) that can bypass the limitations of AFW, Chapter 8 deals with the design and analysis of ExtraFW. The *extra* in its name refers to the pair of gradients involved per iteration, whose merit is to enable a *prediction-correction* (PC) type of update. Though the idea of using two gradients to perform PC updates originates from projection-based algorithms, such as ExtraGradient [164] and Mirror-Prox [165, 166, 167], leveraging PC updates in FW type algorithms for faster rates is novel.

1.4 Bibliographic Notes

The research presented in this dissertation is based on joint work with several co-authors described below.

Chapter 3 is based on the joint work with Longshaokan Wang, Mina Georgieva, Paulo Machado, Abinaya Ulagappa, Safwan Ahmed, Yan Lu, Arjun Bakshi, and Farhad Ghassemi. Chapter 2 is based on the joint work with Savana Ammons, Vera Mikyoung Hur, Ryan L. Sriver, and Zhizhen Zhao. Chpater 4 is based on the joint work with Huozhi Zhou, Bingcong Li, Lav R. Varshney, and Zhizhen Zhao. Chapter 5 is based on the joint work with Bingcong Li, Huozhi Zhou, Georgios, B. Giannakis, Lav R. Varshney, and Zhizhen Zhao. Chapter 6 is based on the joint work with Zhizhen Zhao. Chapter 7 is based on the joint work with Bingcong Li and Georgios B. Giannakis. Chapter 8 is based on the joint work with Bicong Li, Georgios B.

Giannakis, and Zhizhen Zhao.

1.4.1 Excluded Research

Several works are excluded from this dissertation to keep it succinct and coherent. The excluded research includes:

- Work on combinatorial bandits [77].
- Work on 2D tomography from unknown view angles [168].

Part I

Sequential Prediction and Decision Making

CHAPTER 2

CONVOLUTIONAL GRU NETWORK FOR SEASONAL PREDICTION OF THE EL NIÑO-SOUTHERN OSCILLATION

2.1 Preliminaries

2.1.1 ENSO Region Prediction Problem

We address the ENSO region prediction problems which involves predicting future SST map sequences within the ENSO region of the Pacific, given previously observed gridded SST maps of the Pacific region. Suppose that SST maps of the Pacific region are sampled and averaged monthly on a grid of size $M \times N$, representing an SST map of the Pacific region as a matrix in $\mathbb{R}^{M \times N}$ for a specific month. As monthly records of SST maps of the Pacific region are accumulated, a sequence of such matrices is obtained, $\tilde{\boldsymbol{X}}_1, \ldots, \tilde{\boldsymbol{X}}_t, \ldots (\in \mathbb{R}^{M \times N})$, where t denotes a specific month. Given the previous J-month (referred to as the condition range) observed SST maps of the Pacific region, including the current one, represented as $\tilde{\boldsymbol{X}}_{t-J+1:t} \in \mathbb{R}^{J \times M \times N}$. The ENSO region spatio-temporal sequence prediction problem at month t aims to predict the most likely K-month (referred to as the prediction range) future SST maps within the ENSO region. These predicted maps are denoted as $\hat{\boldsymbol{Y}}_{t+1}, \ldots, \hat{\boldsymbol{Y}}_{t+K} (\in \mathbb{R}^{M \times N})$, abbreviated as $\hat{\boldsymbol{Y}}_{t+1:t+K} \in \mathbb{R}^{K \times M \times N}$. Formally, the problem can be stated as follows:

$$\hat{\boldsymbol{Y}}_{t+1:t+K} = \underset{\boldsymbol{Y}_{t+1:t+K}}{\arg\max}\, \mathbb{P}\left(\boldsymbol{Y}_{t+1:t+K} \,\middle|\, \tilde{\boldsymbol{X}}_{t-J+1:t}\right). \tag{2.1}$$

The extent of the ENSO region for the predicted maps can cover the entire Pacific region or any other region within the Pacific, depending on the downstream task, for example, the south Pacific decadal oscillation [169]. In real-world applications, SST maps of the Pacific region are typically sampled and averaged monthly on a latitude-longitude grid of a specific resolution, such as 1° × 1° per latitude-longitude grid cell, and the prediction range spans 12 and 24 months.

The ENSO region prediction problem encompasses a series of downstream tasks, such as predicting Niño indices. It holds potential applications in other climate-related features such as fire weather and drought indices [18, 170].

2.1.2 Models for Spatio-Temporal Sequence Prediction

There exists a range of approaches for spatio-temporal sequence prediction, including ML and traditional statistical models. We categorize these models into autoregressive models and statistical models, highlighting their applicability to weather and climate related tasks.

Autoregressive Models

Autoregressive models have found widespread usage in time series prediction problems, including RNNs such as vanilla RNN [171], LSTM [36], and GRU [172]. One multivariate variant of general-purpose RNN, known as fully-connected RNN (FC-RNN) [38, 173], is among the earliest models employed for the spatio-temporal sequence prediction, which takes vectorized inputs (spatio maps). The main equations for FC-LSTM can be summarized as follows:

$$\begin{aligned}
i_t &= \sigma_i(W_{ix}x_t + W_{ih}h_{t-1} + b_i), && \bullet\ \text{Input gate}\\
f_t &= \sigma_f(W_{fx}x_t + W_{fh}h_{t-1} + b_f), && \bullet\ \text{Forget gate}\\
o_t &= \sigma_o(W_{ox}x_t + W_{oh}h_{t-1} + b_o), && \bullet\ \text{Output gate}\\
\tilde{c}_t &= \sigma_{\tilde{c}}(W_{\tilde{c}x}x_t + W_{\tilde{c}h}h_{t-1} + b_{\tilde{c}}), && \bullet\ \text{New memory cell}\\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, && \bullet\ \text{Final memory cell}\\
h_t &= o_t \odot \sigma_h(c_t), && \bullet\ \text{Hidden state}
\end{aligned}$$

where $\odot$ represents the element-wise product (Hardmard product), and $\sigma(\cdot)$ is either the sigmoid or tanh function.

However, FC-LSTM has limitations in efficiently capturing spatial correlations. To overcome this drawback, ConvLSTM [38] is introduced, which incorporates 2-D convolutional layers within an LSTM cell. ConvLSTM has been further enhanced with various variants and successors such as TrajGRU [39], CDNA [174], PredRNN [175]. Another kind of autoregressive models for spatio-temporal sequence prediction is based on transformers [176, 177]. Additionally, autoregressive models have been combined with other techniques, such as graph neural networks [178, 179] and generative models [180], leading to significant achievements in short to medium-range weather prediction and other spatio-temporal sequence prediction applications.

Traditional Statistical Climate Models

Statistical climate models employ statistical techniques tailored for climate-related data analysis and prediction. Examples include LIM [29, 30] and KAF [28, 33, 34].

LIM assumes that the dynamics of a system can be described by a linear stochastic differential equation of the form:

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{B}\boldsymbol{x} + \xi.$$

Here $\boldsymbol{x}(t)$ represents the state of the system at time t, $\boldsymbol{B}$ is a time-independent operator, and ξ is stationary white noise. For stationary statistics, $\boldsymbol{B}$ must be dissipative, meaning that its eigenvalues have negative real parts, and

$$\boldsymbol{C}(\tau) = \boldsymbol{G}(\tau)\boldsymbol{C}(0) \quad \text{and} \quad \boldsymbol{G}(\tau) = \exp(\boldsymbol{B}\tau),$$

where $\boldsymbol{C}(0)$ and $\boldsymbol{C}(\tau)$ are covariances of $\boldsymbol{x}$ at lags 0 and τ, respectively. In prediction problems, $\boldsymbol{G}(\tau)\boldsymbol{x}(t)$ represents the best linear prediction of the state at time $t+\tau$, given the state at time t. The matrices $\boldsymbol{B}$ and $\boldsymbol{G}$ can then be determined as $\boldsymbol{B} = \tau^{-1} \ln\left(\boldsymbol{C}(\tau)C(0)^{-1}\right)$.

KAF is a generalization of AF [26, 32], incorporating both nonlinear kernel methods and operator-theoretic ergodic theory [181]. By establishing a rigorous connection with Koopman operator theory [182] for dynamical systems, KAF can generate statistically optimal predictions as conditional expectations. KAF is particularly useful when dealing with noisy and partially observed data during prediction initialization.

These models form foundations for the spatio-temporal sequence prediction and have been applied to various weather and climate problems. However, one should note that i) LIM is not capable of capturing nonlinear dynamics with the ENSO region, ii) KAF is usually applied to Niño index prediction task, and is not capable of predicting the spatial pattern within the ENSO region. Those limitations of statistical models motivate the proposal of the ConvGRU network.

2.2 Methodology

We now propose our ConvGRU network, which is inspired by and modified from the ConvGRU model [39, 40], for the ENSO region spatio-temporal sequence prediction. The ConvGRU network incorporates 2D convolutional layers in both input-to-(hidden) state and (hidden) state-to-(hidden) state transitions within a ConvGRU cell. This modification offers several advantages over the FC-GRU cell, efficiently capturing spatial correlations of SST maps and reducing the number of network parameters. The ConvGRU network is composed of multiple stacked ConvGRU cells and follows an encoder-decoder Seq2Seq structure. During the training process, samples are generated from fixed-length windows with different starting points.

2.2.1 Convolutional GRU cell

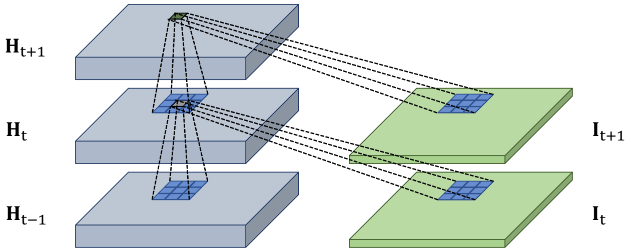


Figure 2.1: Illustration of 2D convolutional layers within a ConvGRU cell. Convolutional layers are applied to update gate, reset gate, and new memory cell (see (2.2)).

If we were to tackle the ENSO region spatio-temporal sequence prediction problem in (2.1) using a network built with FC-GRU cells, we would need to vectorize inputs and hidden states before performing matrix multiplication. These steps are essentially equivalent to fully connected layers in a neural network. However, the vectorization and matrix multiplication steps are not required when using ConvGRU cells. Instead, a ConvGRU cell employs 2D convolutional layers which offers several advantages, including extracting meaningful spatial correlation features, reducing the number of network parameters, and speeding up the training process.

The equations for the ConvGRU cell can be expressed as follows:

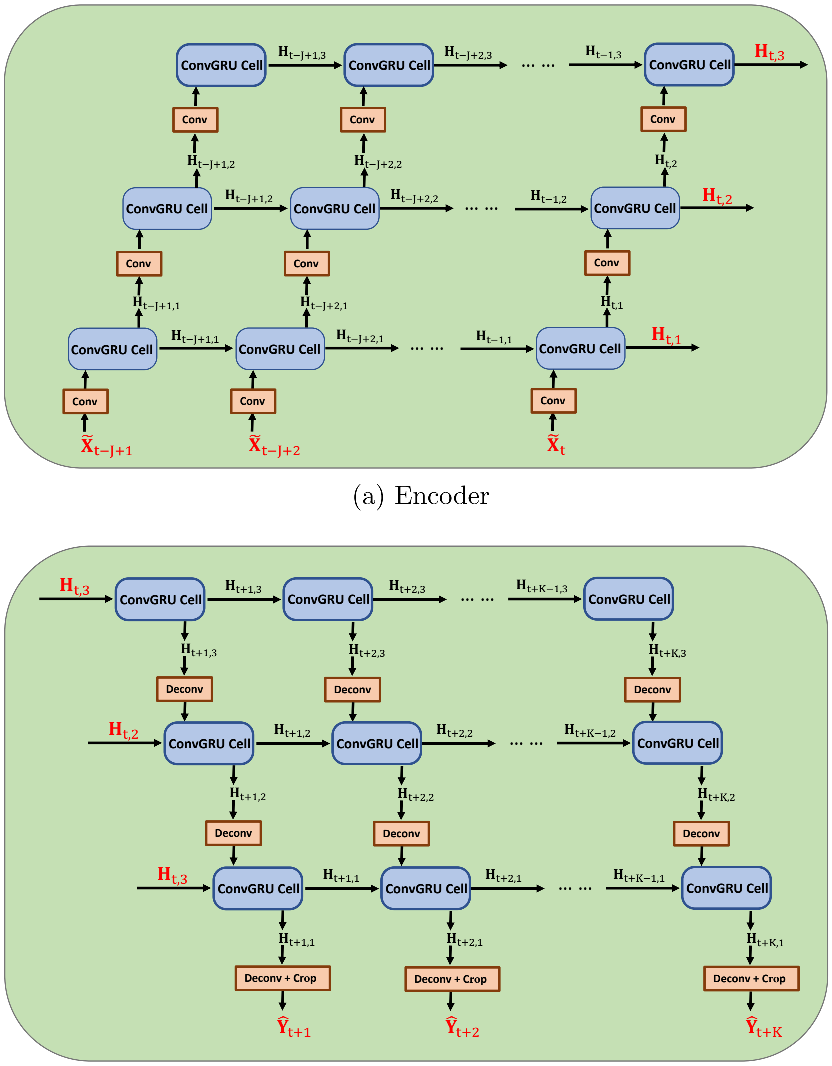
$$
\begin{aligned}
z_t &= \sigma_z(\boldsymbol{W}_{zx} * \boldsymbol{I}_t + \boldsymbol{W}_{zh} * \boldsymbol{H}_{t-1} + b_z), && \bullet\ \text{Update gate} \\
r_t &= \sigma_r(\boldsymbol{W}_{rx} * \boldsymbol{I}_t + \boldsymbol{W}_{rh} * \boldsymbol{H}_{t-1} + b_r), && \bullet\ \text{Reset gate} \\
\tilde{\boldsymbol{H}}_t &= \sigma_{\tilde{h}}(\boldsymbol{W}_{\tilde{h}x} * \boldsymbol{I}_t + \boldsymbol{W}_{\tilde{h}rh} * (r_t \odot \boldsymbol{H}_{t-1}) + b_{\tilde{h}}), && \bullet\ \text{New memory cell} \\
\boldsymbol{H}_t &= (1 - z_t) \odot \tilde{\boldsymbol{H}}_t + z_t \odot \boldsymbol{H}_{t-1}, && \bullet\ \text{Hidden state}
\end{aligned}
\tag{2.2}
$$

where $*$ represents the convolution operator. Here, input $\boldsymbol{I}_t$, hidden state $\boldsymbol{H}_t$, update gate z_t, reset gate r_t, and new memory cell $\tilde{\boldsymbol{H}}_t$ are all 3D tensors, with the last two dimensions representing the spatial dimensions (rows and columns). Figure 2.1 illustrates the application of 2D convolutional layers within a ConvGRU cell for both the input-to-(hidden) state and (hidden) state-to-(hidden) state transitions. This allows the future hidden state in a specific grid cell to extract relevant information locally from its neighboring inputs and past hidden states. The size of the neighbors considered by a grid cell is determined by the size of the convolutional kernel. A large kernel is recommended for fast-evolving spatio-temporal sequences, while a small kernel is more suitable for slow-varying sequences.

2.2.2 Encoder-Decoder Seq2Seq Structure

We utilize ConvGRU cells in (2.2) as a key component to construct our ConvGRU network for ENSO region spatio-temporal sequence prediction. We recognize this as a Seq2Seq learning problem in (2.1) that can be effectively addressed using the encoder-decoder Seq2Seq structure [172, 183].

Figure 2.2 illustrates the architecture of the ConvGRU network for a 3-layer example, where the number of layers can be adjusted based on performance considerations, such as RMSE. The ConvGRU network consists of



(a) Encoder

(b) Decoder

Figure 2.2: Three-layer ConvGRU network, where the initial hidden states of the decoder are copied from the last hidden states of the encoder. (a) Encoder architecture utilizing ConvGRU cells and 2D convolutional layers. (b) Decoder architecture constructed with ConvGRU cells and 2D deconvolutional layer.

two main parts: the multi-layer encoder and the multi-layer decoder.

The encoder, depicted in Figure 2.2a, utilizes ConvGRU cells and 2D convolutional layers. Each layer's hidden states are initialized as an all 0-tensor and updated using inputs and previous hidden states, following (2.2). The first layer takes SST maps of the Pacific region as inputs, while subsequent layers receive output hidden states from the previous layer. Convolutional layers are applied before each ConvGRU cell to adjust the number of the input channels and the size of the input spatial dimensions, enhancing feature extraction.

The decoder, depicted in Figure 2.2b, consists of ConvGRU cells and 2D deconvolutional layers. A crucial step that connects the encoder and the decoder is that hidden states of each layer in the decoder are copied from the last output hidden states of the corresponding layer in the encoder. The decoder architecture is similar to the encoder, but the flow direction of hidden states among layers is reversed. This enables the adoption of the 2D deconvolutional layer, which is the reverse operation of 2D convolutional layer. They ensure that inputs and network parameters in each decoder layer's ConvGRU cells are consistent with those in the encoder, so that the last output hidden states from the encoder can be utilized by the decoder. For outputs of the first layer of the decoder, the grid is cropped to the ENSO region.

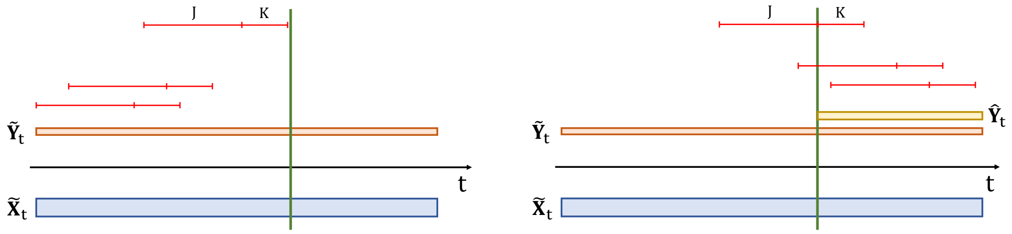
The encoder-decoder Seq2Seq structure of the ConvGRU network can be interpreted as follows: the encoder compresses the input SST maps of the Pacific region into hidden states across all layers, while the decoder unfolds hidden states from the encoder to generate predictions for the ENSO region. Consequently, the ConvGRU network approximates the problem stated in (2.1) as:

$$\begin{aligned}\hat{\boldsymbol{Y}}_{t+1:t+K} &= \underset{\boldsymbol{Y}_{t+1:t+K}}{\arg\max}\ \mathbb{P}\left(\boldsymbol{Y}_{t+1:t+K} \,\middle|\, \tilde{\boldsymbol{X}}_{t-J+1:t}\right) \\ &\approx \underset{\boldsymbol{Y}_{t+1:t+K}}{\arg\max}\ \mathbb{P}\left(\boldsymbol{Y}_{t+1:t+K} \,\middle|\, f_{\text{ENC}}\left(\tilde{\boldsymbol{X}}_{t-J+1:t} \,\middle|\, \boldsymbol{W}_{\text{ENC}}\right)\right) \\ &\approx g_{\text{DEC}}\left(f_{\text{ENC}}\left(\tilde{\boldsymbol{X}}_{t-J+1:t} \,\middle|\, \boldsymbol{W}_{\text{ENC}}\right) \,\middle|\, \boldsymbol{W}_{\text{DEC}}\right),\end{aligned} \tag{2.3}$$

where $f_{\text{ENC}}(\cdot|\boldsymbol{W}_{\text{ENC}})$ and $g_{\text{DEC}}(\cdot|\boldsymbol{W}_{\text{DEC}})$ represent the encoder and decoder, respectively, with network parameters $\boldsymbol{W}_{\text{ENC}}$ and $\boldsymbol{W}_{\text{DEC}}$.

2.2.3 Training Process

Given a training dataset consisting of SST maps, denoted as $\{(\tilde{\boldsymbol{X}}_t, \tilde{\boldsymbol{Y}}_t)\}_{t=1}^{T}$, of the Pacific and ENSO regions, respectively, network parameters $\boldsymbol{W}^*_{\text{ENC}}$ and $\boldsymbol{W}^*_{\text{DEC}}$ of the encoder and the decoder can be learned by minimizing the difference between the predicted sequence $\hat{\boldsymbol{Y}}_{t+1:t+K}$ and the ground truth



(a) Training

(b) Testing

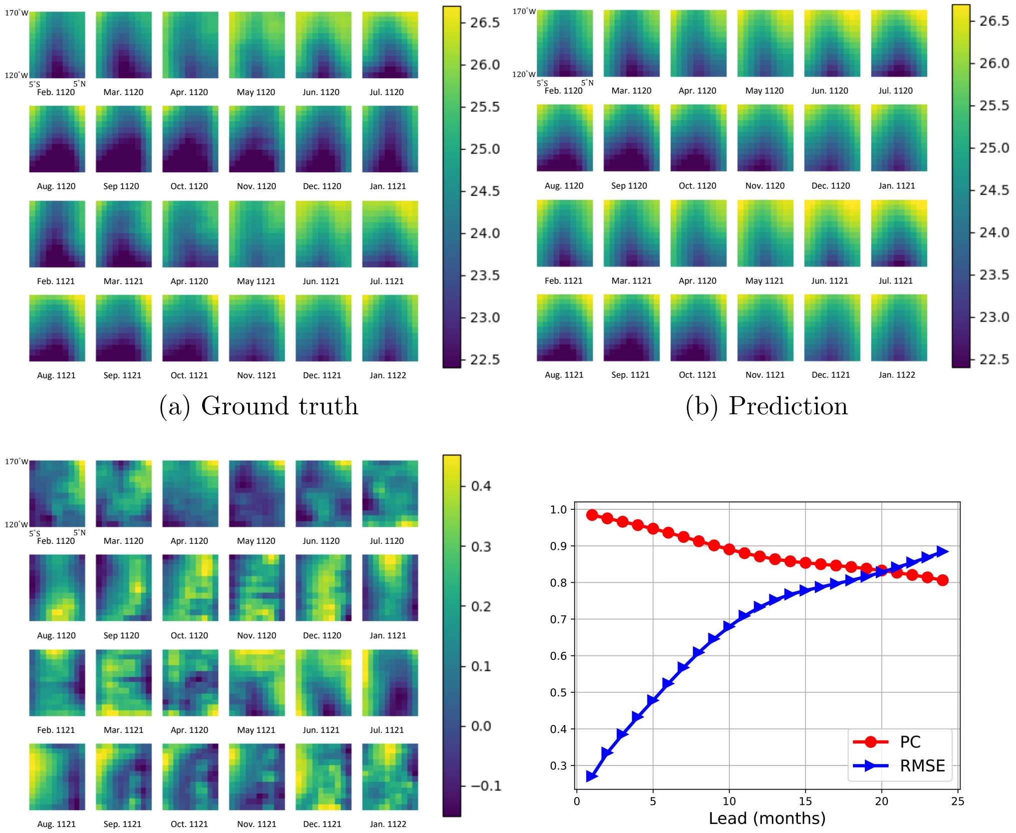
Figure 2.3: Data setup for training and testing. The green vertical lines divide the entire dataset into the training data and the testing data. The performance metric is evaluated to the right of the green line, and no data in this part is used in training. (a) Data setup during the training process. The red lines depict the training windows of $\{(\tilde{\boldsymbol{X}}_t, \tilde{\boldsymbol{Y}}_t)\}_{t=1}^{T}$, where the left part represents the condition range ($t - J + 1$ to t for some t), and the right part represents the prediction range ($t + 1$ to $t + K$). Note that all training windows are to the left of the green line. (b) During the testing process, the prediction range is strictly to the right of the green line.

sequence $\tilde{\boldsymbol{Y}}_{t+1:t+K}$. The optimization process can be described as follows:

$$\begin{aligned} &\boldsymbol{W}^*_{\text{ENC}}, \boldsymbol{W}^*_{\text{DEC}} \\ &= \underset{\boldsymbol{W}_{\text{ENC}}, \boldsymbol{W}_{\text{DEC}}}{\arg\min} \sum_{t=J}^{T-K} \mathcal{L}\left(\tilde{\boldsymbol{Y}}_{t+1:t+K}, g_{\text{DEC}}\left(f_{\text{ENC}}\left(\tilde{\boldsymbol{X}}_{t-J+1:t} \;\middle|\; \boldsymbol{W}_{\text{ENC}}\right) \;\middle|\; \boldsymbol{W}_{\text{DEC}}\right)\right) \\ &= \underset{\boldsymbol{W}_{\text{ENC}}, \boldsymbol{W}_{\text{DEC}}}{\arg\min} \sum_{t=J}^{T-K} \mathcal{L}\left(\tilde{\boldsymbol{Y}}_{t+1:t+K}, \hat{\boldsymbol{Y}}_{t+1:t+K}\right), \end{aligned} \tag{2.4}$$

where the loss function $\mathcal{L}$ is the MSE loss. The optimization problem in (2.4) can be solved using stochastic gradient descent algorithms, such as Adam [184] and Adagrad [185].

During the training process, multiple training windows (instances) of length $J + K$ are generated from the training dataset, each with different start points. The condition (J) and prediction (K) ranges remain fixed for all training windows. For instance, if the training dataset spans from month 1 to month 10000, training windows can be created with t in (2.4) ranging from J to $10000 - K$. Figure 2.3a illustrates the generation of training windows. Once the ConvGRU network is trained, it can be evaluated on a testing dataset $\{(\tilde{\boldsymbol{X}}_t, \tilde{\boldsymbol{Y}}_t)\}_{t=T+1}^{T+F}$, as depicted in Figure 2.3b.



(a) Ground truth

(b) Prediction

(c) Diff.

(d) RMSE and PC

Figure 2.4: Performance on the ENSO region spatio-temporal sequence prediction task. (a): Sample ground truth of the ENSO region starting from February 1120. (b): Sample prediction of the ENSO region starting from February 1120. (c): Sample difference between the ground truth and prediction of the ENSO region start from February 1120. (d): RMSE per grid cell and PC as a function of lead time computed in the testing period.

2.3 Results and Discussion

2.3.1 Experimental Setup

Experiment results and discussions of the ConvGRU network on various global climate simulation and reanalysis datasets are presented. The datasets used consist of two SST simulation datasets and one air temperature reanalysis dataset. The performance of the ConvGRU network is evaluated on the ENSO region spatio-temporal sequence prediction task for the SST datasets. Additionally, the performance is compared with several existing models on a downstream task of predicting the Niño 3.4 index, which is calculated based on the aforementioned ENSO region spatio-temporal sequence prediction task. The ConvGRU network is also evaluated on the spatio-temporal sequence prediction task for the air temperature dataset, which covers almost 2/3 of the global surface.

For numerical experiments, the ConvGRU network is implemented using PyTorch [186]. The experiments are conducted on a Linux server equipped with a single GPU, either NVIDIA GeForce GTX 1080Ti or NVIDIA RTX A6000[1].

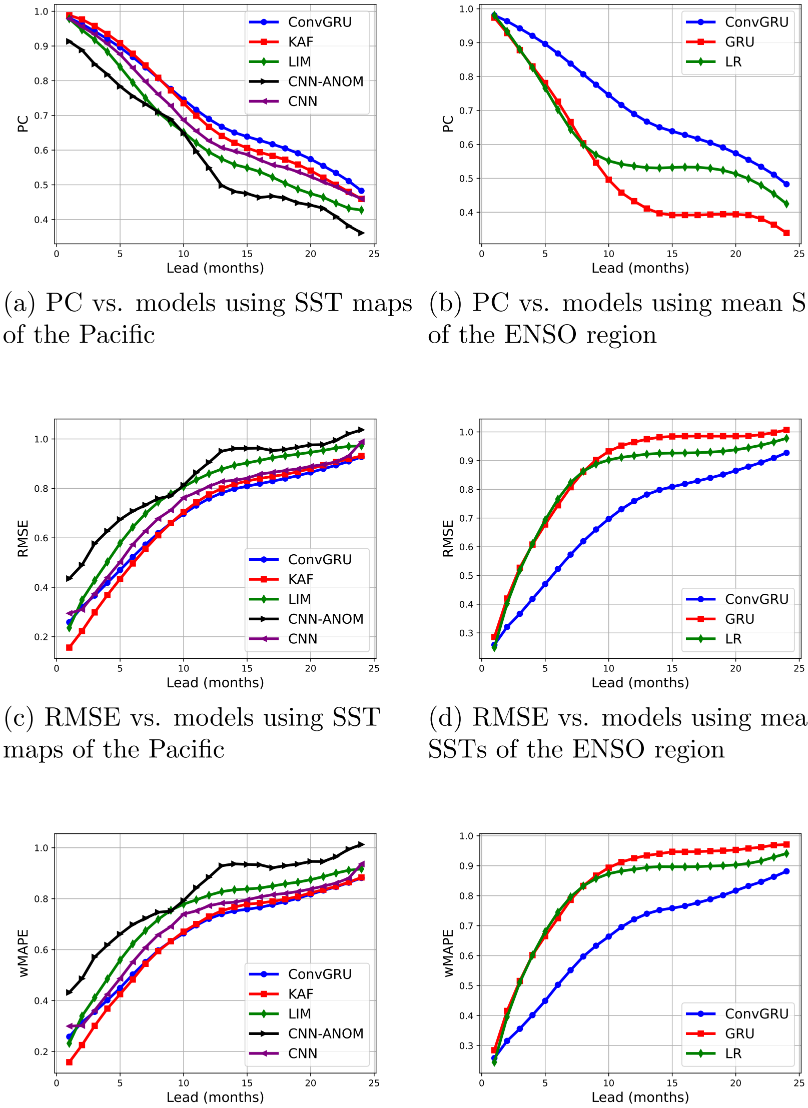
2.3.2 The CCSM4 Simulation Dataset

The CCSM4 dataset is a modeled SST dataset derived from a 1300-year, pre-industrial control integration of the community climate system model version 4 (CCSM4) [187]. The dataset is sampled and averaged monthly on the model's native ocean grid with a normal resolution of approximately $1^\circ \times 0.5^\circ$ (longitude-latitude). The SST maps of the Pacific and ENSO regions are extracted from specific longitude-latitude boxes. The Pacific region covers 16°E-56°W and 69°S-32°N (256×256 grid), while the ENSO region covers 170°-120°W and 5°S-5°N (45×38 grid). To reduce computational complexity and GPU memory usage, the SST maps of the Pacific and ENSO regions are down-sampled to 64×64 and 12×10 grids, respectively. The CCSM4 dataset is split into disjoint training and testing data periods, with year 1-1099 allocated for training and year 1100-1300 for testing.

For experiments on the CCSM4 dataset, a 3-layer ConvGRU network is implemented. The condition range (J) and the prediction range (K) are set to 48 and 24 months, respectively.

Figure 2.4 illustrates the performance of the ConvGRU network on the ENSO region spatio-temporal sequence prediction task. Figures 2.4a, 2.4b, and 2.4c include a sample comparison, starting from February 1120, between the ground truth and the network's prediction for the ENSO region. The patterns in both the ground truth and the prediction exhibit high similarity. Figure 2.4d presents the prediction skill assessed using RMSE and PC metrics over the entire testing period as a function of lead time. The RMSE

[1] The codes and detailed information about the datasets can be found at the following public GitHub repository: `https://github.com/LingdaWang/ConvGRU_ENSO_Forecast`.



(a) PC vs. models using SST maps of the Pacific

(b) PC vs. models using mean SSTs of the ENSO region

(c) RMSE vs. models using SST maps of the Pacific

(d) RMSE vs. models using mean SSTs of the ENSO region

(e) wMAPE vs. models using SST maps of the Pacific

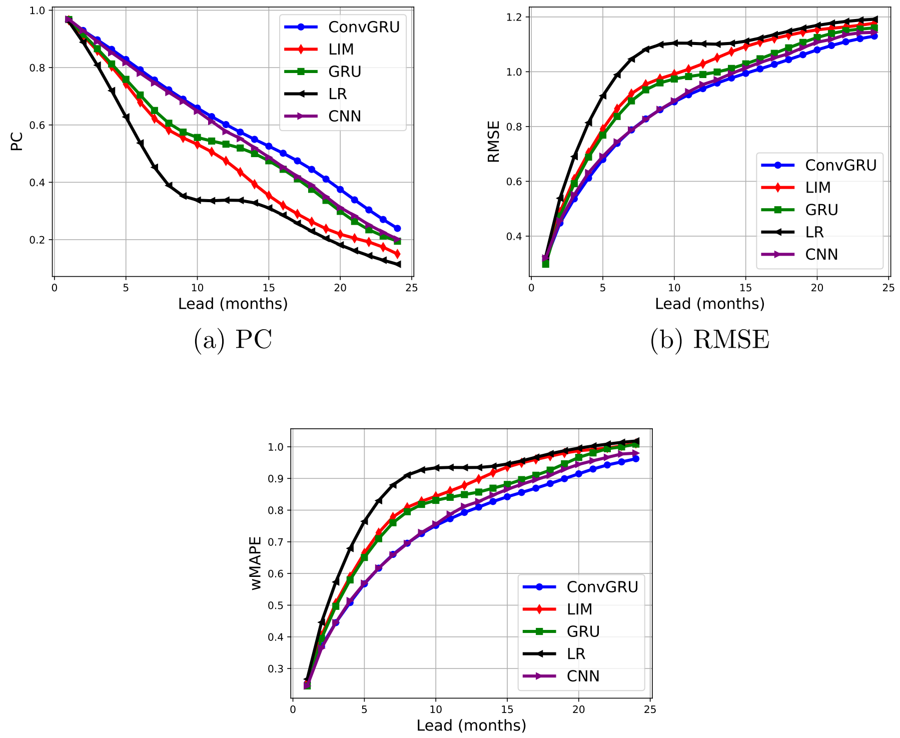
(f) wMAPE vs. models using mean SSTs of the ENSO region

Figure 2.5: Performance of the ConvGRU network against other models on predicting the Niño 3.4 index in the CCSM4 dataset during 1100-1300. (a) PC, (c) RMSE, and (e) wMAPE, respectively, as a function of lead time, compared to KAF, LIM, and CNNs. (b) PC, (d) RMSE, and (f) wMAPE, respectively, as a function of lead time, compared to Seq2Seq with GRU and LR.

values are averaged over all possible start points in the testing data split and grid cells of the ENSO region. For PC, ground truths and predictions are vectorized and concatenated over all possible start points in the testing data split. The RMSE and PC results demonstrate the high correlation and low error characteristics of predictions generated by the ConvGRU network.

Next, the performance of the ConvGRU network is compared with existing models for predicting the Niño 3.4 index in the CCSM4 dataset. Here Niño 3.4 indices mean SST anomalies relative to monthly climatology (average SST) of the ENSO region. The models selected for comparison include KAF [28, 33, 34], LIM [28, 29, 30], CNN [35], Seq2Seq with GRU [172, 183], and LR. KAF, LIM, and CNN utilize SST maps of the Pacific region as input (predictor) variables, while Seq2Seq with GRU and LR use mean SSTs of the ENSO region. The CNN model implementation is based on the Nature paper [35], where three convolution layers and two max pooling layers are included. For inputs of CNN models, CNN and CNN-ANOM utilize the original SST maps and SST anomaly maps respectively, since the paper suggests SST anomaly maps. Seq2Seq with GRU is implemented using the DeepAR model [43] from the GluonTS package [47], with a 1-layer GRU network with a 20-dimensional hidden state, and the condition and prediction ranges (J and K) the same as the ConvGRU network. For LR, $K = 24$ separate models are trained for lead months 1-24, using $J = 48$ months lagged mean SSTs of the ENSO region (including the current month) as input features.

Figure 2.5 illustrates the performance of the ConvGRU network compared against other models in predicting the Niño 3.4 index in the CCSM4 dataset over the testing period of 1100-1300, with the training period from 1 to 1099. A threshold of PC = 0.6 is commonly used to differentiate useful from non-useful predictions [28]. The comparison demonstrates that although the performance of the ConvGRU network deteriorates with longer lead times, it consistently outperforms the competing models in terms of PC, RMSE, and wMAPE, particularly in the long-term prediction range. When considering the useful prediction range using the PC threshold of 0.6, the ConvGRU network achieves the longest useful range of 18-19 months, surpassing KAF, CNN, LIM, CNN-ANOM, Seq2Seq with GRU, and LR by 3-4, 5, 6-7, 7-8, 10-11, and 10-11 months, respectively.



(a) PC

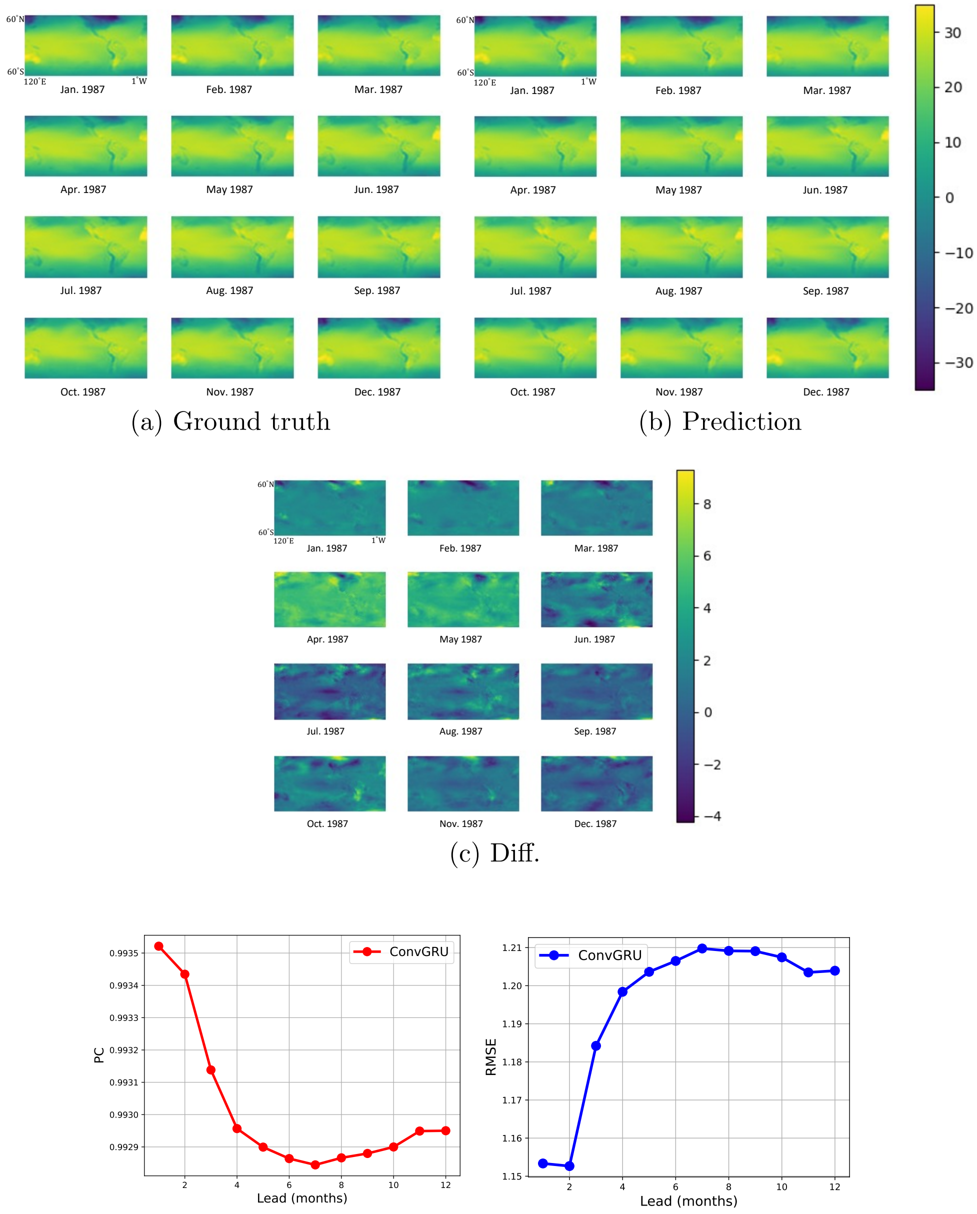
(b) RMSE

(c) wMAPE

Figure 2.6: Performance averaged over 30 ensembles of the ConvGRU network against other models on predicting the Niño 3.4 index in the NOAA-GDFL-SPEAR dataset during 2051-2100. (a) PC, (b) RMSE, and (c) wMAPE, respectively, as a function of lead time, compared to LIM, CNN, Seq2Seq with GRU, and LR.

2.3.3 The NOAA-GDFL-SPEAR Simulation Dataset

The NOAA-GDFL-SPEAR dataset used in the numerical experiment is a simulated monthly averaged SST dataset with a nominal resolution of $1^\circ \times 1^\circ$ (longitude-latitude) from the GFDL SPEAR large ensembles. This includes 30-member ensembles of climate change simulations covering the period 1921-2100 using the SPEAR-MED climate model [188]. The simulations are forced with historical radiative forcings from 1921 to 2014 and SSP5-8.5 projected radiative forcings [189, 190] from 2015 to 2100. The SST maps of the Pacific and ENSO region are extracted from the longitude-latitude boxes 150°E-82°W, 69°S-59°N (128×128 grid), and 170°-120°W, 5°S-5°N (50×10 grid), respectively. Similar to the previous comparison, the SST maps in both regions are down-sampled to 64×64 and 25×5 grids, reducing computational



(a) Ground truth

(b) Prediction

(c) Diff.

(d) PC

(e) RMSE

Figure 2.7: Performance on the air temperature spatio-temporal sequence prediction task. (a) Sample ground truth starting from January 1987. (b) Sample prediction starting from January 1987. (c) Sample difference between ground truth and prediction starting from January 1987. (d) PC as a function of lead time computed in the testing period. (e) RMSE per grid cell as a function of lead time computed in the testing period.

complexity and GPU memory usage.

The NOAA-GDFL-SPEAR dataset in each ensemble is divided into disjoint training and testing data splits, with the training period covering years 1921-2050 and the testing period covering years 2051-2100. For experiments on this dataset, a 3-layer ConvGRU network is implemented. The ConvGRU network is trained using data from all 30 ensembles that end on or before the year 2050 and then tested on data from all 30 ensembles starting from the year 2051. The condition range (J) and the prediction range (K) are 48 and 24 months, respectively.

Similar to the previous comparison, the performance of the ConvGRU network is compared to several existing models for predicting the Niño 3.4 index in the NOAA-GDFL-SPEAR dataset. The selected models for comparison include LIM, CNN, Seq2Seq with GRU, and LR. For LIM and LR, separate models are trained for each ensemble in the dataset, since the NOAA-GDFL-SPEAR dataset contains data from 30 ensembles, and the metrics computed in the testing period are averaged over all ensembles. The CNN model follows the same setting as that for the CCSM4 dataset. Seq2Seq with GRU utilizes the DeepAR model to handle multiple time series with a single model, and the detailed and fine-tuned settings remain the same in the CCSM4 dataset.

Figure 2.6 presents the performance averaged over 30 ensembles of the ConvGRU network compared against other models in predicting the Niño 3.4 index in the NOAA-GDFL-SPEAR dataset over the testing period of 2051-2100, using the training period of 1921-2051. The results of the experiments demonstrate that the ConvGRU network significantly outperforms the competing models in terms of PC, RMSE, and wMAPE, with the longest useful range of 12 months, surpassing CNN, LIM, Seq2Seq with GRU, and LR by 1, 4-5, 4, and 7 months, respectively.

2.3.4 The NOAA-CIRES Air Temperature Reanalysis Dataset

The NOAA-CIRES air temperature dataset used in this experiment is a monthly ensemble mean air temperature dataset at the 2m level with a nominal resolution of approximately 2° × 2° (longitude-latitude). It is from the NOAA-CIRES 20th-century reanalysis version 2c [191, 192], which provides a comprehensive global atmospheric circulation dataset spanning the years 1850-2014. For the NOAA-CIRES air temperature dataset, we aim to demonstrate that the ConvGRU network is capable of accurately predicting other climate and atmospheric spatio-temporal sequence beyond the ENSO region. For this experiment, the target region is the longitude-latitude box 120°E – 1°W, 60°N – 60°S (128×64 grid), covering almost two-thirds of the total global surface.

The NOAA-CIRES air temperature dataset is divided into disjoint training and testing periods, with the training period covering the years 1851-1980 and testing period covering 1981-2014. In this experiment, a 3-layer ConvGRU network is implemented. The condition range (J) and the prediction range (K) are set to 24 and 12 months, respectively.

Figure 2.7 illustrates the performance of the ConvGRU network on the air temperature spatio-temporal sequence prediction task. Figures 2.7a, 2.7b, and 2.7c presents a sample comparison starting from January 1987 between the ground truth and the prediction, revealing a very similar pattern in both the ground truth and the prediction. Figures 2.7d and 2.7e present the prediction skill assessed using RMSE and PC, similar to Fig. 2.4, over the entire testing data split and as a function of lead time. The PC result demonstrates a significantly high correlation of over 99% between the ground truth and the prediction within a prediction range of 12 months. The RMSE ranges from approximately 1.1 ℃ to 1.2 ℃.

CHAPTER 3

ROBUST NONPARAMETRIC DISTRIBUTION FORECAST WITH BACKTEST-BASED BOOTSTRAP AND ADAPTIVE RESIDUAL SELECTION

3.1 Methodology

The proposed DF framework is composed of a backtester, a residual selector, and a PF model, as depicted in Figure 3.1. To summarize how it works: During the training phase: 1. Backtest [45] is performed on the training data with the PF model to build a collection of predictive residuals (Figure 3.2); for covariates that need to be estimated for future time points (e.g., future price of a product), their values can be sampled from estimated distributions during backtest to account for the uncertainty in covariates. 2. The residual selector is pre-specified or learned from the training data as a set of rules or a separate machine learning model that selects the most relevant subset of predictive residuals given a future data point based on their meta information. 3. Lastly, the PF model is trained on the entire training data. During the forecasting phase: 1. For each future data point of interest, the trained PF model generates the PF. 2. The residual selector selects a subset of residuals. 3. Lastly, random samples of residuals are drawn from the subset and applied to the PF to obtain multiple bootstrap forecasts that provide the empirical distribution, and sample quantiles of the bootstrap forecasts provide the quantile forecasts for arbitrary target quantiles. Essentially, we use the empirical distribution of the selected predictive residuals from backtest to estimate the distribution of the future predictive residuals and thus the distribution of the future response variable.

3.1.1 Backtesting

Let $\mathcal{D} = \{(\mathbf{X}_i^t, Y_i^t)\}_{i=1,2,\ldots,n}^{t=s_i,s_i+1,\ldots,d_i}$ be the training data, where $\mathbf{X}_i^t$ is the matrix of covariates at time t, Y_i^t is the response variable at time t, s_i is the

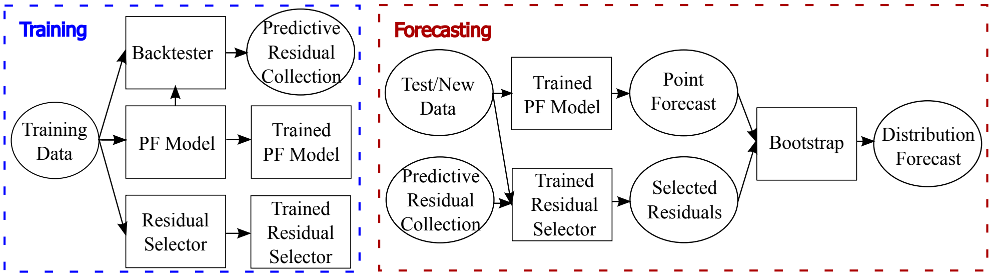


Figure 3.1: Overview of the proposed DF framework. The backtester generates a collection of predictive residuals; the residual selector selects a subset of residuals for each future data point; the bootstrapping step combines the PF and selected residuals to generate the DF.

first time point, and d_i is the last time point for time series i. For a non-parametric distribution forecast, it suffices to estimate the conditional quantiles $\widehat{Q}_{Y_i^{d_i+k_i}|Y_i^{s_i:d_i},\mathbf{X}_i^{s_i:d_i},\mathbf{X}_i^{(d_i+1):(d_i+k_i)}}(\tau)$ for arbitrary target quantile $\tau \in (0,1)$, where k_i is the number of time points into the future. Backtest is essentially a move-forward cross-validation that preserves the order in time for time series data, where the test split is always further in time than the training split. Let the backtest start time and step size be a and l respectively. For each split point $j = a, a+l, a+2l, \ldots, \max_i(d_i) - 1$, the training data are divided into a training split $\mathcal{A}_j = \{(\mathbf{X}_i^t, Y_i^t) \in \mathcal{D} \,|\, t \leq j\}$ and a test split $\mathcal{B}_j = \{(\mathbf{X}_i^t, Y_i^t) \in \mathcal{D} \,|\, t > j\}$; the PF model $\widehat{f}_j$ is trained on $\mathcal{A}_j$, and predictive residuals are computed as $\{Y_i^t - \widehat{f}_j(Y_i^{s_i:j}, \mathbf{X}_i^{s_i:j}, \mathbf{X}_i^{(j+1):t}) | (\mathbf{X}_i^t, Y_i^t) \in \mathcal{B}_j\}$. This process generates a collection of predictive residuals $\mathcal{E} = \{\varepsilon_{i,j}^t\}_{i,j,t}$. For those covariates that are not available in the future and need to be estimated, we can use their historic estimates, sample from their estimated distributions, or add simulated noise to create $\widetilde{\mathbf{X}}_i^t$ to replace $\mathbf{X}_i^t$ during backtest to account for uncertainty in covariates.

3.1.2 Selecting Residuals

Common PF models typically assume that residuals are i.i.d. and independent from the covariates and the PF itself [42]. However, such assumptions do not always hold in practice for the predictive residuals. For example, the variance of residuals can increase as we forecast further into the future or as the magnitude of PF increases. To relax the commonly imposed independence assumption between residuals and covariates (or more generally

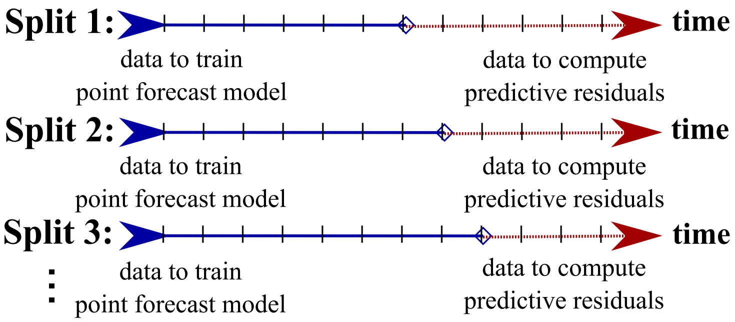


Figure 3.2: Illustration of building a collection of predictive residuals with backtest. The training split is used to train the PF model, and the test split is used to compute the predictive residuals.

any meta information which can include the PF or other variables not in the original covariates), an adaptive residual selector can be learned from the training data to select a subset of residuals based on the meta information of the predictive residuals from backtest and the future data point, $\widehat{g}(\mathcal{E}, \mathcal{M}, \mathcal{M}^{\text{future}})$, so that the selected residuals are conditionally i.i.d.. The residual selector should ideally be based on the meta information that has a non-negligible impact on the predictive residuals. We mention two options for learning the residual selector here: 1. Compute distance correlation (which can detect both linear and non-linear dependence) [193] between the predictive residuals from backtest and their corresponding meta information to identify variables with the highest distance correlation to the residuals. Then design rules (e.g., set simple thresholds) around these variables to select residuals that have a different distribution from the distribution of the entire collection of residuals, which can be verified by the Kolmogorov-Smirnov test [194]. Note that if the residual selector has no impact, the selected residuals should have the same distribution as the entire collection. 2. Fit a machine learning model, such as a regression decision tree, to predict residuals from their meta information, then apply the model to the meta information of future data points to select the corresponding residuals. The performance of this model can also be used to check dependence between meta information and residuals – if the residuals are already independent from the meta information pre-selection, then the model should perform poorly.

3.1.3 Bootstrapping

We describe two formulae of generating bootstrap forecasts, *backtest-additive* (BA) and *backtest-multiplicative* (BM). They can be applied to either iterative or direct PF models (an iterative model recursively consumes the forecast from the previous time point to forecast for the next, whereas a direct model generates forecast for a future time point directly from covariates [195]). For BA, to generate bootstrap forecasts for the next time point $d_i + 1$, after obtaining the PF $\widehat{Y}_i^{d_i+1} = \widehat{f}(Y_i^{s_i:d_i}, \mathbf{X}_i^{s_i:d_i}, \mathbf{X}_i^{d_i+1})$ and the selected predictive residuals from backtest $\mathcal{G} = \widehat{g}(\mathcal{E}, \mathcal{M}, \mathcal{M}_i^{d_i+1})$, random samples are drawn from the selected residuals $\varepsilon_b \in \mathcal{G}$ for $b = 1, 2, \ldots, B$, then the bootstrap forecasts are given by $\widehat{Y}_{i,b,\text{Add.}}^{d_i+1} = \widehat{Y}_i^{d_i+1} + \varepsilon_b$. Quantile forecasts are obtained by taking sample quantiles of the bootstrap forecasts. Generalizing to arbitrary future time point $d_i + k_i$, for an iterative PF model, bootstrap forecasts are recursively generated for the next time point until $d_i + k_i$; for a direct PF model, the calculation remains the same as 1-step forecast with $d_i + 1$ replaced by $d_i + k_i$. Note that for a direct PF model, quantile forecasts can be obtained by skipping the residual sampling step and adding the sample quantiles of the selected residuals to the PF. The formula for BA is similar to the existing approach to bootstrapping predictive residuals [196, 197] (while the backtest and residual selection steps are novel in BA). The performance of BA can degrade if the variance of residuals increases with the magnitude of the PF or if the magnitude of the future PF is very different from the magnitude of the response variable seen during backtest. Hence we also propose BM which scales the residuals based on the PF: After obtaining the PF and the selected residuals in the same way as BA, the error ratios are computed by dividing the extracted residuals over their corresponding forecast (or response variable) during backtest, $\mathcal{R} = \{\varepsilon_{h,j}^t / \widehat{Y}_{h,j}^t \mid \varepsilon_{h,j}^t \in \mathcal{G}\}$; then the bootstrap forecasts for the next time point are given by sampling $r_b \in \mathcal{R}$ and $\widehat{Y}_{i,b,\text{Multi.}}^{d_i+1} = \widehat{Y}_i^{d_i+1} \cdot (1 + r_b)$. The rest remains the same.

3.1.4 Practical Considerations

Both the backtest step and the residual selection step can be efficiently parallelized across multiple CPU's/GPU's. The backtest step requires multiple model training, but it is more efficient than the previous *delete*-$\mathbf{x}^t$ approach of bootstrapping predictive residuals [196, 197] and can be done offline at a lower frequency than updating the PF model. The only computational overhead during inference time is the (optional) residual selection given the PF, so the additional latency of obtaining DF is negligible. Furthermore, once a residual collection from backtest is built, quantile forecast for any target quantile can be generated without re-running backtest or retraining the PF model, whereas DF methods that explicitly minimize quantile loss typically require the target quantile to be specified before model training. The backtest-based methods are also relatively interpretable: They retain the interpretability of the underlying PF model if the PF model is interpretable; even with a less interpretable PF model, one can check the predictive residual distribution and model performance on the test split (and model coefficients if applicable) during the backtest step to help identify which data points or covariates tend to contribute to large predictive residuals and whether the model has systematic bias during out-of-sample forecasting. The choices of bootstrap formulae (BA vs BM), denominator of error ratios (backtest forecast vs observed response variable), residual selector variation, and PF model can be tuned as hyperparameters.

3.2 Experiments

Table 3.1: ACE comparison of different bootstrap DF approaches integrated with different PF models.

Bootstrap\PF	Ridge	SVR	RF	NN
FR	0.102(−0%)	0.195(−0%)	0.207(−0%)	0.176(−0%)
FM	0.095(−7%)	0.218(+12%)	0.171(−17%)	0.125(−29%)
BA	0.069(−32%)	0.065(−67%)	0.055(−73%)	0.077(−56%)
BM	**0.038**(**−63%**)	**0.061**(**−69%**)	**0.027**(**−87%**)	**0.048**(**−73%**)

We conduct experiments on two real-world time-series datasets: an in-house product sales dataset and the M4-hourly competition dataset [46, 47]. The product sales dataset consists of daily sales of 76 products between 01/01/2017 and 01/10/2021 and 147 covariates capturing information on pricing, supply constraints, trend, seasonality, special events, and product attributes. The standard ACE is used to evaluate the DF performance:

The coverage (CO) of quantile forecast $\widehat{Y}^t_{i(\tau)}$ for target quantile τ over the test set $\mathcal{D}_{\text{test}}$ is defined as $\text{CO}(\mathcal{D}_{\text{test}};\tau) = \frac{1}{|\mathcal{D}_{\text{test}}|}\sum_{\mathcal{D}_{\text{test}}} I\{Y^t_i \leq \widehat{Y}^t_{i(\tau)}\}$; and ACE is defined as $\text{ACE}(\mathcal{D}_{\text{test}};\tau) = |\text{CO}(\mathcal{D}_{\text{test}};\tau) - \tau|$. A 100-fold backtest is used for evaluation, which is separate from the backtest used for computing predictive residuals – in each training-test split for evaluation, the latter half of the training split is used to perform a separate backtest to build the predictive residual collection without using information from the test split for a fair evaluation. The reported ACE is averaged across all training-test splits, 24-week forecast horizon for product sales and 48-hour horizon for M4-hourly, and the following range of target quantiles: $\tau = 0.1, 0.2, \ldots, 0.9$. For experiments with deep learning models, the reported ACE is also averaged across 10 trials due to the fluctuation in model performance.

Table 3.2: ACE comparison of backtest-based bootstrap integrated with the median forecast vs the default DF.

DF\Model	QLasso	QGB	DeepAR	DFact	MQCNN	DSSM	TFT
Default	0.188	0.119	0.102	0.098	0.092	0.136	0.067
Median + BA	0.114	0.078	**0.100**	**0.067**	0.078	0.124	**0.058**
Median + BM	**0.039**	**0.036**	0.104	0.070	**0.071**	**0.112**	0.060

Compared to other DF approaches, bootstrap approaches have the advantage of extending any PF model to produce DF, which makes them easy to adopt and able to potentially retain desired properties of the PF model. Thus, the first experiment focuses on comparing the proposed BA and BM against classic bootstrap approaches for DF: bootstrap with fitted residuals (FR) [42] and bootstrap with fitted models (FM) [198, 197]. This experiment is performed on the product sales dataset, as it contains covariates which can accommodate the use of standard Machine Learning models as direct PF models. A variety of PF models are used to assess the bootstrap approaches' robustness to the choice of PF model, including ridge regression [199], support vector regression (SVR) [199], random forest (RF) [199], and neural networks (NNs) [199]. The proposed bootstrap approaches outperform the classic approaches for all PF models (Table 3.1).

The second experiment compares against other SOTA DF approaches, including quantile lasso (QLasso) [199], quantile gradient boosting (QGB) [199], DeepAR [43, 47], Deep Factors (DFact) [200, 47], MQ-CNN [201, 47],

Table 3.3: ACE comparison of backtest-based bootstrap integrated with the median forecast vs the default DF from DeepAR under different pre-specified output distributions.

DF\Output Dist.	Neg. Bin.	Student's t	Normal	Gamma	Laplace	Poisson
Default	0.102	0.192	0.162	**0.138**	0.114	0.134
Median + BA	**0.100**	0.169	0.116	0.157	0.094	0.128
Median + BM	0.104	**0.165**	**0.111**	0.156	**0.088**	**0.125**

deep state space models (DSSM) [44, 47], and temporal fusion transformers (TFT) [202, 47]. Because the bootstrap approaches require an underlying PF model, for a fair comparison we use the median forecast from each of the aforementioned benchmarks as the PF models to be integrated with the backtest-based bootstrap, so they share the same model architecture and hyperparameters. The comparison against QGB and QLasso is performed on the product sales data and the comparison against the deep learning models is performed on the M4-hourly data. The proposed bootstrap approaches integrated with the median forecast outperform the default DF from the benchmarks (Table 3.2).

The third experiment assesses the robustness of the proposed approaches to model assumptions/hyperparameters. DeepAR requires the output distribution to be specified prior to the model learning its parameters. In this experiment, the backtest-based bootstrap approaches integrated with the median forecast are compared against the default DF from DeepAR under a variety of output distribution assumptions on the M4-hourly data. The proposed approaches outperform the default DF in 5 out of 6 distribution settings (Table 3.3).

The median or mean forecast from the bootstrap approaches can be viewed as the updated PF through bootstrap aggregating (Bagging). As an ensemble output, the Bagging PF can be potentially more accurate than the original PF. The fourth experiment evaluates the relative change in mean absolute percentage error (MAPE) of the Bagging PF compared to the original PF on the product sales data. The Bagging PF from the proposed approaches achieves the greatest reduction in MAPE (Table 3.4) for all PF models; i.e., in addition to providing DF, the proposed approaches can also provide more accurate PF. One explanation is that if a PF model has systematic bias during backtest, its predictive residual distribution will reflect such bias, so by design the median forecast will correct for the bias from backtest when bootstrapping DF (Section 3.1.3).

Table 3.4: Relative change in MAPE for Bagging PF compared to the original PF.

Bootstrap\PF Model	Ridge	SVR	RF	NN
FR	+0.8%	+6.5%	+0.2%	+0.7%
FM	+0.4%	+6.6%	−3.8%	+2.6%
BA	−12.3%	−21.0%	**−10.0**%	+1.5%
BM	**−22.1**%	**−31.8**%	−5.3%	**−13.4**%

CHAPTER 4

NEARLY OPTIMAL ALGORITHMS FOR PIECEWISE-STATIONARY CASCADING BANDITS

4.1 Problem Formulation

4.1.1 Cascade Model and Cascading Bandits

CB [6], as a learning variant of CM, depicts the interaction between an agent and a user over a length-T time horizon, in which the user's preference is learned. CM [51] explains the user's behavior in a specific time slot t.

In CM, a user is presented with a K-item ranked list $\mathcal{A}_t := (a_{1,t}, \ldots, a_{K,t}) \in \Pi_K(\mathcal{L})$ from $\mathcal{L}$ at time slot t, where $\mathcal{L} := \{1, 2, \ldots, L\}$ is a ground set containing L items (e.g., web pages or advertisements), and $\Pi_K(\mathcal{L})$ is the set of all K-permutations of the ground set $\mathcal{L}$. CM can be parameterized by an attraction probability vector $\mathbf{w}_t = [\mathbf{w}_t(1), \ldots, \mathbf{w}_t(L)]^\top \in [0,1]^L$. The user browses the list $\mathcal{A}_t$ from the first item a_1 in order, and each item a_k attracts the user to click it with probability $\mathbf{w}_t(a_k)$. The user will stop the process after clicking the first attractive item. In particular, when an item $a_{k,t}$ is clicked, it means that i) items from $a_{1,t}$ to $a_{k-1,t}$ are not attractive to the user, and ii) items $a_{k+1,t}$ to $a_{K,t}$ are not browsed so whether they are attractive to the user is unknown. Clearly, if no item is attractive, the user will browse the whole list and click on nothing.

Building upon CM, a CB can be described by a tuple ($\mathcal{L}$, $\mathcal{T}$, $\{f_{\ell,t}\}_{\ell\in\mathcal{L},t\in\mathcal{T}}$, K), where $\mathcal{T} := \{1, 2, \ldots, T\}$ collects all T time slots. Whether the user is attracted by item ℓ at time slot t is actually a Bernoulli random variable $Z_{\ell,t}$, whose probability mass function (PMF) is $f_{\ell,t}$. As convention, $Z_{\ell,t} = 1$ indicates item ℓ is attractive to the user. We also denote $\mathbf{Z}_t := \{Z_{\ell,t}\}_{\ell\in\mathcal{L}}$ as all the attraction variables of the ground set. Clearly, $\{f_{\ell,t}\}_{\ell\in\mathcal{L},t\in\mathcal{T}}$ are parameterized by the attraction probability vectors $\{\mathbf{w}_t\}_{t\in\mathcal{T}}$, which are unknown to the agent. Since CB is designed for stationary environments, the attraction probability vector $\mathbf{w}_t$ is time-invariant, and thus can be further simplified as $\mathbf{w}$. CB poses a mild assumption on $\{f_{\ell,t}\}_{\ell\in\mathcal{L},t\in\mathcal{T}}$ for simplicity.

Assumption 4.1. *The attraction distributions $\{f_{\ell,t}\}_{\ell\in\mathcal{L},t\in\mathcal{T}}$ are independent both across items and time slots.*

Per slot t, the agent recommends a list of K items $\mathcal{A}_t$ to the user based on the feedback from the user up to time slot $t-1$. The feedback at time slot t refers to the index of the clicked item, given by

$$F_t = \begin{cases} \emptyset, & \text{if no click,} \\ \arg\min_k\{1 \le k \le K : Z_{a_{k,t},t} = 1\}, & \text{otherwise.} \end{cases}$$

After the user browses the list and follows the protocol described by CM, the agent observes the feedback F_t. Along with F_t is a zero-one reward indicating whether there is a click

$$r\left(\mathcal{A}_t, \mathbf{Z}_t\right) = 1 - \prod_{k=1}^{K}\left(1 - Z_{a_{k,t},t}\right), \tag{4.1}$$

where $r\left(\mathcal{A}_t, \mathbf{Z}_t\right) = 0$ if $F_t = \emptyset$. Then, this process proceeds to time slot $t+1$. The goal of the agent is to maximize the expected cumulative reward over the whole time horizon $\mathcal{T}$. Noticing that $Z_{\ell,t}$s are independent, the expected reward at time slot t can be computed as $\mathbb{E}\left[r\left(\mathcal{A}_t, \mathbf{Z}_t\right)\right] = r\left(\mathcal{A}_t, \mathbf{w}\right)$. The optimal list $\mathcal{A}^*$ remains the same for all time slots, which is the list containing the K most attractive items.

4.1.2 Piecewise-Stationary Cascading Bandits

The stationarity assumption on CB limits its applicability for real world applications, as users tend to change their preferences as time goes on [53]. This fact leads to piecewise-stationary CB. Consider a piecewise-stationary CB with N segments, where the attraction probabilities of items remain identical per segment. Mathematically, N can be written as

$$N = 1 + \sum_{t=1}^{T-1}\mathbb{I}\{\exists \ell \in \mathcal{L} \text{ s.t. } f_{\ell,t} \neq f_{\ell,t+1}\}, \tag{4.2}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function, and a change-point is the time slot t that satisfies $\exists \ell \in \mathcal{L}$ s.t. $f_{\ell,t} \neq f_{\ell,t+1}$. Hence it is clear that there are $N-1$ change-points in the piecewise-stationary CB considered. These change-points are denoted by $\nu_1, \ldots, \nu_{N-1}$ in a chronological manner. Specifically, $\nu_0 = 0$ and $\nu_N = T$ are introduced for consistency. For the ith piecewise-stationary segment $t \in [\nu_{i-1}+1, \nu_i]$, f_ℓ^i and $\mathbf{w}^i(\ell)$ denote the attraction distribution and the expected attraction of item ℓ, respectively, which are again unknown to the agent. Attraction probability vector $\mathbf{w}^i = [\mathbf{w}^i(1), \ldots, \mathbf{w}^i(L)]^\top$ is introduced to collect $\mathbf{w}^i(\ell)$s.

In a piecewise-stationary CB, agent interactions are the same as CB. The agent's policy can be evaluated by its expected cumulative reward, or equivalently its expected cumulative regret:

$$\mathcal{R}(T) = \mathbb{E}\left[\sum_{t=1}^{T} R\left(\mathcal{A}_t, \mathbf{w}_t, \mathbf{Z}_t\right)\right], \tag{4.3}$$

where the expectation $\mathbb{E}[\cdot]$ is taken with respect to a sequence of $\mathbf{Z}_t$ and the corresponding $\mathcal{A}_t$. Here, $R(\mathcal{A}_t, \mathbf{w}_t, \mathbf{Z}_t) = r(\mathcal{A}_t^*, \mathbf{w}_t) - r(\mathcal{A}_t, \mathbf{Z}_t)$ is the regret at time slot t with

$$\mathcal{A}_t^* = \operatorname*{arg\,max}_{\mathcal{A}_t \in \Pi_K(\mathcal{L})} r\left(\mathcal{A}_t, \mathbf{w}_t\right)$$

being the optimal list that maximizes the expected reward at time slot t. The regret defined in (4.3) is also known as switching regret, which is widely adopted in piecewise-stationary bandit [203, 57, 61, 60, 62]. Since switching regret is measured with respect to the optimal piecewise-stationary policy, the optimal list $\mathcal{A}_t^*$ for each time slot is no longer time-invariant. This leads to a much harder algorithm design problem since the non-stationary environment should be properly coped with.

4.2 Algorithms

This section presents adaptive approaches for piecewise-stationary CB using an efficient change-point detector.

Algorithm 4.1: GLRT Change-Point Detector: GLRT($X_1, \ldots, X_n; \delta$)

```
Require: observations $X_1, \ldots, X_n$ and confidence level $\delta$
Compute the GLR statistic GLR($n$) according to (4.4) and the threshold $\beta(n, \delta)$ according to (4.5)
if GLR($n$) $\geq \beta(n, \delta)$ then
    Return True
else
    Return False
end if
```

4.2.1 Generalized Likelihood Ratio Test

As the adaptive approach is adopted, a brief introduction about change-point detection is given in this section. Sequential change-point detection is of fundamental importance in statistical sequential analysis, see e.g., [204, 205, 206, 207, 208, 209]. However, the aforementioned approaches typically rely on the knowledge of either pre-change or post-change distribution, rendering barriers for the applicability in piecewise-stationary CB.

In general, with pre-change and post-change distributions unknown, developing algorithms with provable guarantees is challenging. Our solution relies on the GLRT that is summarized under Algorithm 4.1. Compared with existing change-point detection methods that have provable guarantees [61, 60], advantages of GLRT are threefold: i) *Fewer tunable parameters.* The only required parameter for GLRT is the confidence level of change-point detection δ, while CUSUM [61] and CMSW [60] have three and two parameters to be manually tuned, respectively. ii) *Less prior knowledge needed.* GLRT does not require the information on the smallest magnitude among the change-points, which is essential for CUSUM. iii) *Better performance.* The GLRT is more efficient than CUSUM and CMSW in the averaged detection time. As shown in the numerical experiments in Example 4.1, GLRT has approximately 20% and 50% improvement over CUSUM and CMSW, respectively.

Next, the GLRT is formally introduced. Suppose we have a sequence of Bernoulli random variables $\{X_t\}_{t=1}^n$ and aim to determine if a change-point exists as fast as we can. This problem can be formulated as a parametric sequential test of the following two hypotheses:

$$\begin{aligned}
&\mathcal{H}_0 : \exists \mu_0 : X_1, \ldots, X_n \overset{\text{i.i.d}}{\sim} \text{Bern}(\mu_0), \\
&\mathcal{H}_1 : \exists \mu_0 \neq \mu_1,\ \tau \in [1, n-1] : X_1, \ldots, X_\tau \overset{\text{i.i.d}}{\sim} \text{Bern}(\mu_0) \\
&\quad \text{and } X_{\tau+1}, \ldots, X_n \overset{\text{i.i.d}}{\sim} \text{Bern}(\mu_1),
\end{aligned}$$

where Bern(μ) is the Bernoulli distribution with mean μ. The GLR statistic is

$$\text{GLR}(n) = \sup_{s \in [1, n-1]} [s \times \text{KL}\left(\hat{\mu}_{1:s}, \hat{\mu}_{1:n}\right) + (n-s) \times \text{KL}\left(\hat{\mu}_{s+1:n}, \hat{\mu}_{1:n}\right)], \quad (4.4)$$

where $\hat{\mu}_{s:s'}$ is the empirical mean of observations from X_s to $X_{s'}$, and KL(x, y) is Kullback–Leibler (KL) divergence of two Bernoulli distributions,

$$\text{KL}(x, y) = x \log\left(\frac{x}{y}\right) + (1-x) \log\left(\frac{1-x}{1-y}\right).$$

By comparing GLR(n) in (4.4) with the threshold $\beta(t, \delta)$, one can decide whether a change-point appears for a length n sequence, where

$$\beta(t, \delta) = 2\mathcal{G}\left(\frac{\log(3t\sqrt{t}/\delta)}{2}\right) + 6\log(1 + \log t), \quad (4.5)$$

and $\mathcal{G}(\cdot)$ has the same definition as that in [210, (13)]. The choice of δ is influences the sensitivity of the GLRT. For example, a larger δ makes the GLRT response faster to a change-point, but increases the probability of false alarm.

The efficiency of a change-point detector for a length n sequence is evaluated via its detection time,

$$\tau = \inf\{t \leq n : \text{GLR}(t) \geq \beta(t, \delta)\}.$$

To better understand the performance of GLRT against CUSUM and CMSW, it is instructive to use an example.

Example 4.1 (Efficiency of GLRT). *Consider a sequence of Bernoulli random variables* $\{X_t\}_{t=1}^n$ *with* $n = 4000$*, where* $X_1, \cdots, X_{2000}$ *are generated from* Bern(0.2) *and the remaining ones are generated from* Bern(0.8)*, as*

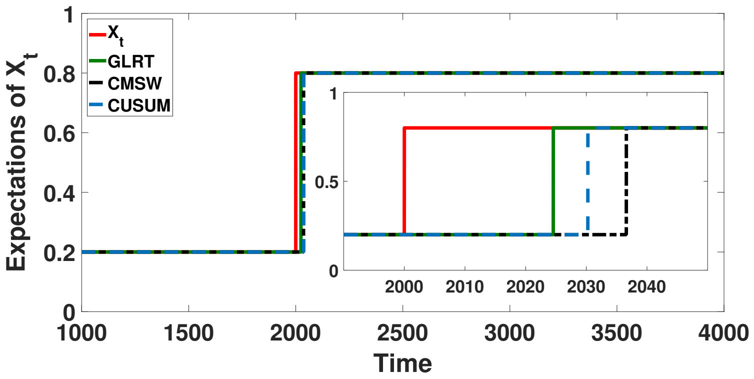


Figure 4.1: Expectations of X_t's with $n = 4000$ and expected detection time of GLRT, CUSUM, and CMSW.

shown in Figure 4.1 (red line). By setting $\delta = 1/n$ *for GLRT and choosing parameters of CUSUM and CMSW as recommended in [61] and [60], the average detection times after 100 Monte Carlo trials are* 2024.55 ± 6.8451 *(GLRT, green line),* 2030.25 ± 6.74 *(CUSUM, blue line), and* 2045.59 ± 4.48 *(CMSW, black line), respectively. In a nutshell, GLRT improves about* 20% *over CUSUM and* 50% *over CMSW.*

4.2.2 The GLRT Based CB Algorithms

Leveraging GLRT as the change-point detector, the proposed algorithms, `GLRT-CascadeUCB` and `GLRT-CascadeKL-UCB`, are presented in Algorithm 4.2. On a high level, three phases comprise the proposed algorithms.

- *Phase 1*: The forced uniform exploration to ensure that sufficient samples are gathered for all items to perform the GLRT detection (Algorithm 4.1).
- *Phase 2*: The upper confidence bound (UCB) based exploration (UCB or Kullback-Leibler UCB(KL-UCB)) to learn the optimal list on each piecewise-stationary segment.
- *Phase 3*: The GLRT change-point detection (Algorithm 4.1) to monitor if global restart should be triggered.

Besides the time horizon $\mathcal{T}$, the ground set $\mathcal{L}$, the number of items in list K, the proposed algorithms only require two parameters p and δ as inputs. The probability p is used to control the portion of uniform exploration in Phase 1, and it appears also in other bandit algorithms for piecewise-stationary environments [60, 61]. While the confidence level δ is the only parameter required by GLRT. Hence, the proposed algorithms are more practical compared with existing algorithms [60, 61], since: i) no prior knowledge on change-point-dependent parameter is needed; ii) fewer parameters are required. The choices of δ and p will be clear in Section 4.3.

In Algorithm 4.2, we denote the last detection time as τ. From slot τ to current slot, let n_l denote the number of observations for ℓth item, and $\hat{\mathbf{w}}(\ell)$ its corresponding sample mean. The algorithm determines whether to perform a uniform exploration or a UCB-based exploration depending on whether line 4 of Algorithm 4.2 is satisfied, which ensures the fraction of time slots performing the uniform exploration phase is about p. If the uniform exploration is triggered, the first item in the recommended list $\mathcal{A}_t$ will be item $a := (t-\tau) \mod \lfloor \frac{L}{p} \rfloor$, and the remaining items in the list are chosen uniformly at random (line 5), which ensures item a will be observed by the user. If UCB-based exploration is adopted at time slot t, the algorithms will choose K items (line 7) with K largest UCB indices,

$$\mathcal{A}_t = \arg \max_{\mathcal{A} \in \Pi_K(\mathcal{L})} r\left(\mathcal{A}, \text{UCB or } \text{UCB}_{\text{KL}}\right), \tag{4.6}$$

which will be defined in (4.7) and (4.8). By recommending the list $\mathcal{A}_t$ and observing the user's feedback F_t (line 9), we update the statistics (line 11) and perform the GLRT detection (line 12). If a change-point is detected, we set $n_\ell = 0$ for all $\ell \in \mathcal{L}$, and $\tau = t$ (line 13). Finally, the UCB indices of each item are computed as follows (line 18),

$$\text{UCB}(\ell) = \hat{\mathbf{w}}(\ell) + \sqrt{\frac{3\log(t-\tau)}{2n_\ell}}, \tag{4.7}$$

$$\text{UCB}_{\text{KL}}(\ell) = \max\{q \in [\hat{\mathbf{w}}(\ell), 1] : n_\ell \times \text{KL}(\hat{\mathbf{w}}(\ell), q) \leq g(t-\tau)\}, \tag{4.8}$$

where $g(t) = \log t + 3\log\log t$, and $\hat{\mathbf{w}}(\ell) = \frac{1}{n_\ell}\sum_{n=1}^{n_\ell} X_{\ell,n}$. Notice that (4.7) is the UCB indices of `GLRT-CascadeUCB`, and (4.8) is the UCB indices of `GLRT-CascadeKL-UCB`. For the intuitions behind, we refer the readers to [211,

Algorithm 4.2: The `GLRT-CascadeUCB` and `GLRT-CascadeKL-UCB` Algorithms

```
Require: The time horizon $\mathcal{T}$, the ground set $\mathcal{L}$, $K$, exploration probability $p > 0$, and
    confidence level $\delta > 0$
Initialization: $\tau \leftarrow 0$ and $n_\ell \leftarrow 0, \forall \ell \in \mathcal{L}$
for all $t = 1, 2, \ldots, T$ do
    $a \leftarrow (t - \tau) \mod \lfloor \frac{L}{p} \rfloor$
    if $a \leq L$ then
        Choose $\mathcal{A}_t$ such that $a_{1,t} \leftarrow a$ and $a_{2,t}, \ldots, a_{K,t}$ are chosen uniformly at random
    else
        Compute the list $\mathcal{A}_t$ follows (4.6)
    end if
    Recommend the list $\mathcal{A}_t$ to user, and observe feedback $F_t$
    for all $k = 1, \ldots, F_t$ do
        $\ell \leftarrow a_{k,t}$, $n_\ell \leftarrow n_\ell + 1$, $X_{\ell,n_\ell} \leftarrow \mathbb{I}\{F_t = k\}$,
        and $\hat{\mathbf{w}}(\ell) = \frac{1}{n_\ell} \sum_{n=1}^{n_\ell} X_{\ell,n}$
        if GLRT$(X_{\ell,1}, \ldots, X_{\ell,n_\ell}; \delta)$ = True then
            $n_\ell \leftarrow 0, \forall \ell \in \mathcal{L}$, and $\tau \leftarrow t$
        end if
    end for
    for $\ell = 1, \cdots, L$ do
        if $n_\ell \neq 0$ then
            Compute UCB$(\ell)$ according to (4.7) for GLRT-CascadeUCB or UCB$_{\mathrm{KL}}(\ell)$
                according to (4.8) for GLRT-CascadeKL-UCB
        end if
    end for
end for
```

Proof of Theorem 1] and [212, Proof of Theorem 2].

4.3 Theoretical Results

The theoretical guarantees of the proposed algorithms, `GLRT-CascadeUCB` and `GLRT-CascadeKL-UCB`, will be derived in this section. Specifically, the upper bounds on the regret of both proposed algorithms are developed in Sections 4.3.1 and 4.3.2. A minimax regret lower bound for piecewise-stationary CB is established in Section 4.3.3. We further discuss our theoretical findings in Section 4.3.4.

Without loss of generality, for the ith piecewise-stationary segment, the ground set $\mathcal{L}$ is first sorted in decreasing order according to attraction probabilities, that is $\mathbf{w}^i(s_i(1)) \geq \mathbf{w}^i(s_i(2)) \geq \cdots \geq \mathbf{w}^i(s_i(L))$, for all $s_i(\ell) \in \mathcal{L}$. The optimal list at ith segment is thus all the permutations of the list $\mathcal{A}_i^* = \{s_i(1), \ldots, s_i(K)\}$. The item ℓ^* is optimal if $\ell^* \in \{s_i(1), \ldots, s_i(K)\}$, otherwise an item ℓ is called suboptimal. To simplify the exposition, the gap between the attraction probabilities of the suboptimal item ℓ and the optimal item ℓ^* at ith segment is defined as:

$$\Delta^i_{\ell,\ell^*} = \mathbf{w}^i(\ell^*) - \mathbf{w}^i(\ell).$$

Similarly, the largest amplitude change among items at change-point ν_i is defined as

$$\Delta^i_{\text{change}} = \max_{\ell\in\mathcal{L}} \left|\mathbf{w}^{i+1}(\ell) - \mathbf{w}^i(\ell)\right|, \tag{4.9}$$

with $\Delta^0_{\text{change}} = \max_{\ell\in\mathcal{L}} |\mathbf{w}^1(\ell)|$. We have the following assumption for the theoretical analysis.

Assumption 4.2. *Define $d_i = d_i(p,\delta) = \lceil \frac{4L\beta(T,\delta)}{p(\Delta^i_{\text{change}})^2} + \frac{L}{p} \rceil$ and assume $\nu_i - \nu_{i-1} \geq 2\max\{d_i, d_{i-1}\}$, $\forall i = 1,\ldots,N-1$, with $\nu_N - \nu_{N-1} \geq 2d_{N-1}$.*

Note that Assumption 4.2 is standard in a piecewise-stationary environment, and identical or similar assumptions are made in other change-detection based bandit algorithms [61, 60, 62] as well. It requires the length of the piecewise-stationary segment between two change-points to be large enough. Assumption 4.2 guarantees that with high probability, all the change-points are detected within the interval $[\nu_i + 1, \nu_i + d_i]$, which is equivalent to saying all change-points are detected correctly (low probability of false alarm) and quickly (low detection delay). This result is formally stated in Lemma 4.3. In our numerical experiments, the proposed algorithms work well even when Assumption 4.2 does not hold (see Section 4.4).

4.3.1 Regret Upper Bound for `GLRT-CascadeUCB`

Upper bound on the regret of `GLRT-CascadeUCB` is as follows.

Theorem 4.1. *Suppose that Assumptions 4.1 and 4.2 are satisfied,* `GLRT-CascadeUCB` *guarantees*

$$\mathcal{R}(T) \leq \underbrace{\sum_{i=1}^{N} \widetilde{C}_i}_{(a)} + \underbrace{Tp}_{(b)} + \underbrace{\sum_{i=1}^{N-1} d_i}_{(c)} + \underbrace{3NTL\delta}_{(d)},$$

where $\widetilde{C}_i = \sum_{\ell=K+1}^{L} \frac{12}{\Delta^i_{s_i(\ell),s_i(K)}} \log T + \frac{\pi^2}{3} L$.

Theorem 4.1 indicates that the upper bound on the regret of `GLRT-CascadeUCB` is incurred by two types of costs that are further decomposed into four terms. Terms (a) and (b) upper bound the costs of UCB-based exploration and uniform exploration, respectively. The costs incurred by the change-point detection delay and the incorrect detection are bounded by terms (c) and (d). Corollary 4.1 follows directly from Theorem 4.1.

Corollary 4.1. *Let $\Delta_{\text{change}}^{\min} = \min_{i \le N-1} \Delta_{\text{change}}^{i}$ denote the smallest magnitude of any change-point on any item, and $\Delta_{\text{opt}}^{\min} = \min_{i \le N} \Delta_{s_i(K+1), s_i(K)}^{i}$ be the smallest magnitude of a suboptimal gap on any one of the stationary segments. The regret of* `GLRT-CascadeUCB` *is established by choosing $\delta = \frac{1}{T}$ and $p = \sqrt{\frac{NL\log T}{T}}$:*

$$\mathcal{R}(T) = \mathcal{O}\left(\frac{N(L-K)\log T}{\Delta_{\text{opt}}^{\min}} + \frac{\sqrt{NLT\log T}}{\left(\Delta_{\text{change}}^{\min}\right)^2}\right). \tag{4.10}$$

As a direct result of Theorem 4.1, the upper bound on the regret of `GLRT-CascadeUCB` in Corollary 4.1 consists of two terms where the first term is incurred by the UCB-based exploration and the second term is from the change-point detection component. As T becomes larger, the regret is dominated by the cost of the change-point detection component, implying the regret is $\mathcal{O}(\sqrt{NLT\log T}/(\Delta_{\text{change}}^{\min})^2)$. Similar phenomena can also be found in piecewise-stationary MAB [61, 60, 62].

The proof outline of Theorem 4.1 is as follows. We can decompose $\mathcal{R}(T)$ into good events that `GLRT-CascadeUCB` reinitializes the algorithm quickly and precisely after all change-points and bad events that either large detection delays or false alarms happen. We first upper bound the regret of the stationary scenario and the detection delays of good events, respectively. It can be shown that with high probability, all change-points can be detected correctly and quickly, so that the regret incurred by bad events is rather small. By summing up all regrets from good events and bad events, an upper bound on the regret of `GLRT-CascadeUCB` is then developed.

4.3.2 Regret Upper Bound for `GLRT-CascadeKL-UCB`

This section deals with the upper bound on the T-step regret of `GLRT-CascadeKL-UCB`.

Theorem 4.2. *Suppose that Assumptions 4.1 and 4.2 are satisfied,* `GLRT-CascadeKL-UCB` *guarantees*

$$\mathcal{R}(T) \leq \underbrace{T(N-1)(L+1)\delta}_{(a)} + \underbrace{Tp}_{(b)} + \underbrace{\sum_{i=1}^{N-1} d_i}_{(c)} + \underbrace{NK\log\log T + \sum_{i=0}^{N-1} \widetilde{D}_i}_{(d)},$$

where $\widetilde{D}_i$ is a term depending on $\log T$ *and the suboptimal gaps. Detailed expression can be found in* (4.12) *in the Section 4.6.*

Similarly, the upper bound on the regret of `GLRT-CascadeKL-UCB` in Theorem 4.2 can be decomposed into four different terms where (a) is incurred by the incorrect change-point detections, (b) is the cost of the uniform exploration, (c) is incurred by the change-point detection delay, and (d) is the cost of the KL-UCB based exploration.

Corollary 4.2. *Choosing the same δ and p as in Corollary 4.1,* `GLRT-CascadeKL-UCB` *has same order of regret upper bound as* (4.10).

We sketch the proof for Theorem 4.2 as follows, and the detailed proofs are presented in Section 4.6. By defining the events $\mathcal{U}$ and $\mathcal{H}_T$ as the algorithm performing uniform exploration and the change-points can be detected correctly and quickly, we can first bound the cost of uniform exploration $\mathcal{U}$ and cost of incorrect and slow detection of change-points $\overline{\mathcal{H}}_T$. Then, we can divide the regret $\mathcal{R}(T)$ into different piecewise-stationary segments. By bounding the cost of detection delays and the KL-UCB based exploration, the upper bound on regret is thus established.

4.3.3 Minimax Regret Lower Bound

In this section, we derive a minimax regret lower bound for piecewise-stationary CB which is tighter than $\Omega(\sqrt{T})$ proved in [56]. The proof technique is significantly different from [56].

Theorem 4.3. *If $L \geq 3$ and $T \geq MN\frac{(L-1)^2}{L}$, then for any policy, the worst-case regret is at least $\Omega(\sqrt{NLT})$, where $M = {}^{1}\!/\!\log\frac{4}{3}$, and $\Omega(\cdot)$ notation hides a constant factor that is independent of N, L, and T.*

The high-level idea is constructing a randomized hard instance appropriate for the piecewise-stationary CB setting in which per time slot there is only one item with highest click probability, and the click probabilities of remaining items are the same. When the distribution change occurs, the best item changes uniformly at random. For this instance, in order to lower bound the regret, it suffices to upper bound the expected numbers of appearances of the optimal item in the list. We then apply a change of measure technique to upper bound this expectation. One key step is to apply the data processing inequality for KL divergence to upper bound the discrepancy of feedback F_t under change of distribution.

This lower bound is the first characterization involving N, L, and T. It indicates our proposed algorithms are nearly order-optimal within a logarithm factor $\sqrt{\log T}$.

4.3.4 Discussion

Corollaries 4.1 and 4.2 reveal that by properly choosing the confidence level δ and the uniform exploration probability p, the regrets of `GLRT-CascadeUCB` and `GLRT-CascadeKL-UCB` can be upper bounded by

$$\mathcal{R}(T) = \mathcal{O}\left(\sqrt{NLT\log T}\right),$$

where $\mathcal{O}(\cdot)$ notation hides the gap term $\Delta_{\text{change}}^{\min}$ and the lower order term ${}^{N(L-K)\log T}\!/\!{}_{\Delta_{\text{opt}}^{\min}}$. Note that compared to CUSUM in [61, Liu et al., 2018] and CMSW in [60, Cao et al., 2019], the tuning parameters are fewer and do not require the smallest magnitude among the change-points $\Delta_{\text{change}}^{\min}$ as shown in Corollary 4.1. Moreover, parameter δ and p follow simple rules as shown in Corollary 4.1, while complicated parameter tuning steps are required in CUSUM and CMSW.

The upper bounds on the regret of `GLRT-CascadeUCB` and `GLRT-Cascade-KL-UCB` are improved over state-of-the-art algorithms `CascadeDUCB` and `CascadeSWUCB` in [56, Li et al., 2019] either in the dependence on L or both L and

T, as their upper bounds are $\mathcal{O}(L\sqrt{NT}\log T)$ and $\mathcal{O}(L\sqrt{NT\log T})$, respectively. In real-world applications, both L and T can be huge. For example, L and T are in the millions in web search, which reveals the significance of the improved L dependence in our bounds. Compared to recent works on piecewise-stationary MAB [62] and combinatorial MAB (CMAB) [77] that adopt GLRT as the change-point detector, the problem setting considered herein is different. In MAB, only one selected item rather than a list of items is allowed per time slot. Notice that although CMAB [213, 214, 215] or non-stationary CMAB [77] also allow a list of items, they have full feedback on all K items under semi-bandit setting.

4.4 Numerical Experiments

In this section, numerical experiments on both synthetic and real-world datasets are carried out to validate the effectiveness of proposed algorithms. Four baseline algorithms are chosen for comparison, where `CascadeUCB1` [6] and `CascadeKL-UCB` [6] are nearly optimal algorithms to handle stationary CB; while `CascadeDUCB` [56] and `CascadeSWUCB` [56] cope with piecewise-stationary CB through a passively adaptive manner. In addition, two oracle algorithms, `Oracle-CascadeUCB1` and `Oracle-CascadeKL-UCB`, that have access to change-point times are also selected for comparison. In particular, the oracle algorithms restart when a change-point occur. Based on the theoretical analysis by [56], we choose $\xi = 0.5$, $\gamma = 1 - 0.25/\sqrt{T}$ for `CascadeDUCB` and choose $\tau = 2\sqrt{T\log T}$ for `CascadeSWUCB`. For `GLRT-CascadeUCB` and `GLRT-CascadeKL-UCB`, we set $\delta = 1/T$ and $p = 0.1\sqrt{N\log T/T}$.

4.4.1 Synthetic Dataset

In this experiment, let $L = 10$ and $K = 3$. We consider a simulated piecewise-stationary environment setup as follows: i) the expected attractions of the top K items remain constant over the whole time horizon; ii) in each even piecewise-stationary segment, three suboptimal items are chosen randomly and their expected attractions are set to be 0.9; iii) in each odd piecewise-stationary segment, we reset the expected attractions to the initial state. In this experiment, we set the length of each piecewise-stationary segment to

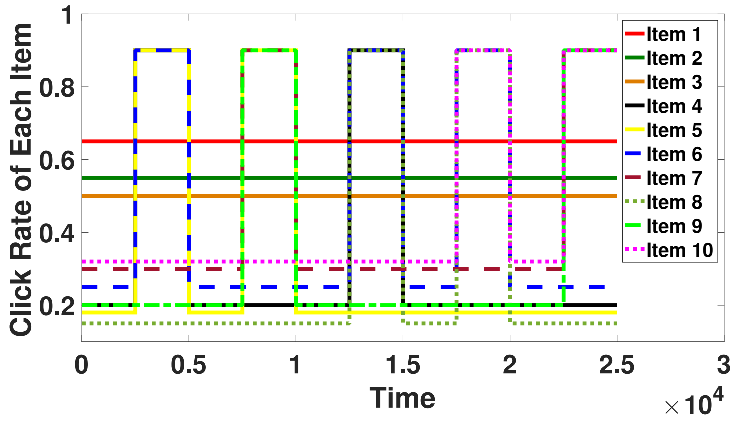


Figure 4.2: Click rate of each item of synthetic dataset with $T = 25000$, $L = 10$ and $N = 10$.

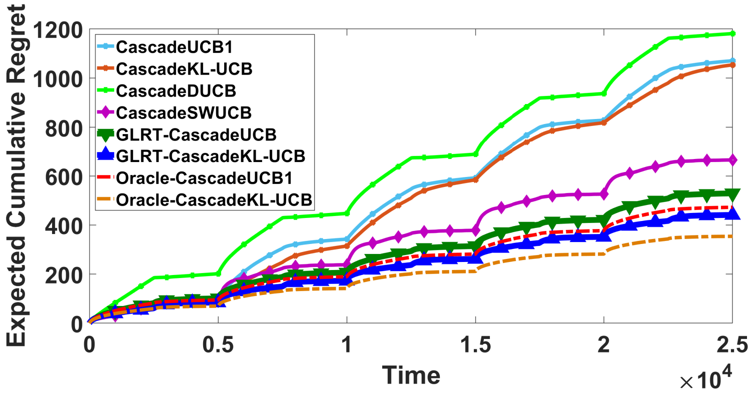


Figure 4.3: Expected cumulative regrets of different algorithms on synthetic dataset.

be 2500 and choose $N = 10$, which is a total of 25000 steps. Figure 4.2 is a detailed depiction of the piecewise-stationary environment.

Figure 4.3 reports the T-step cumulative regrets of all the algorithms by taking the average of the regrets over 100 Monte Carlo simulations. The results show that the proposed `GLRT-CascadeUCB` and `GLRT-CascadeKL-UCB` achieve better performances than other algorithms and are very close to the oracle algorithms. Compared with the best existing algorithm, `GLRT-CascadeUCB` achieves a 20% reduction of the cumulative regret and this fraction is 33% for `GLRT-CascadeKL-UCB`, which is consistent with difference of empirical results between passively adaptive approach and actively adaptive approach in MAB. Notice that although `CascadeDUCB` seems to capture the change-

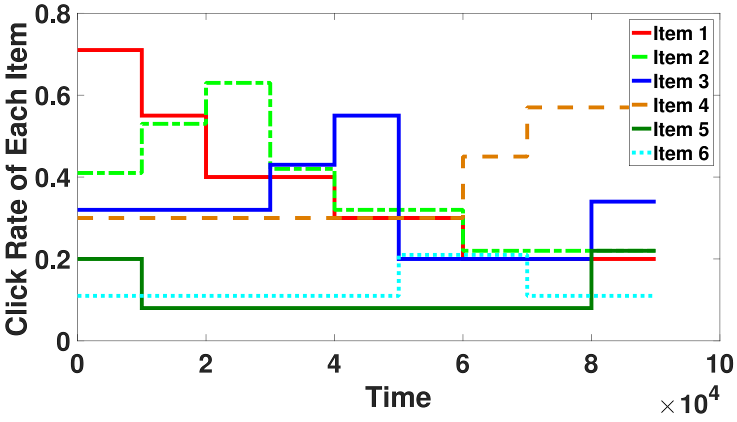


Figure 4.4: Click rate of each item of Yahoo! dataset with $T = 90000$, $L = 6$ and $N = 9$.

points, the performance is even worse than algorithms designed for stationary CB. TThere are two possible reasons: i) The theoretical result shows that `CascadeDUCB` is worse than other algorithms for piecewise-stationary CB by a $\sqrt{\log T}$ factor; ii) the time horizon $\mathcal{T}$ is not long enough. It is worth mentioning that our experiment on this synthetic dataset violates Assumption 4.2, as it would require more than 10^5 time slots for each piecewise-stationary segment. Surprisingly, the proposed algorithms are capable of detecting all the change-points correctly with high probability and sufficiently fast in our experiments.

4.4.2 Yahoo! Dataset

In this section, we adopt the benchmark dataset for the evaluation of bandit algorithms published by Yahoo![1]. This dataset, using binary values to indicate if there is a click or not, contains user click log for news articles displayed in the Featured Tab of the Today Module on Yahoo! [216], where each item corresponds to one article. We pre-process the dataset by adopting the same method as [60, Cao et al., 2019], where $L = 6$, $K = 2$ and $N = 9$. To make the experiment nontrivial, several modifications are applied to the dataset: i) the click rate of each item is enlarged by 10 times; ii) the time horizon

[1] Yahoo! Front Page Today Module User Click Log Dataset on https://webscope.sandbox.yahoo.com

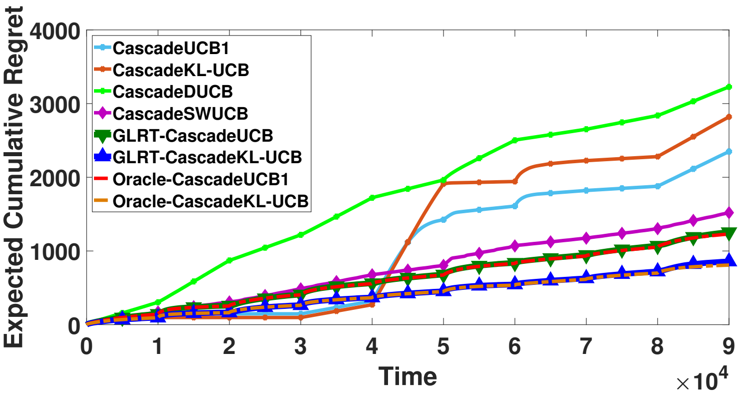


Figure 4.5: Expected cumulative regrets of different algorithms on Yahoo! dataset.

is reduced to $T = 90000$, which is shown in Figure 4.4. Figure 4.5 presents the cumulative regrets of all algorithms by averaging 100 Monte Carlo trials which shows the regrets of our proposed algorithms are just slightly above the oracle algorithms and significantly outperform other algorithms. The reason that algorithms designed for stationarity perform better in the first three segments is the optimal list does not change.

4.5 Proof of Theorem 4.1

4.5.1 Proofs of Auxiliary Lemmas

In this section, we present auxiliary lemmas which are used to prove Theorem 4.1, as well as their proofs. We start by upper bounding the regret under the stationary scenario with $N = 1$, $\nu_0 = 0$, and $\nu_1 = T$.

Lemma 4.1. *Under stationary scenario ($N = 1$), the regret of* `GLRT-CascadeUCB` *is upper bounded as*

$$\mathcal{R}(T) \leq T\mathbb{P}\left(\tau_1 \leq T\right) + pT + \widetilde{C}_1,$$

where τ_1 is the first detection time.

Proof of Lemma 4.1. Denote $R_t := R\left(\mathcal{A}_t, \mathbf{w}_t, \mathbf{Z}_t\right)$ as the regret of the learning algorithm at time slot t, where $\mathcal{A}_t$ is the recommended list at time slot t

and $\mathbf{w}_t$ is the associated expected attraction vector at time slot t. By further denoting as τ_1 the first change-point detection time of the Bernoulli GLRT, the regret of `GLRT-CascadeUCB` can be decomposed as:

$$\begin{aligned}\mathcal{R}(T) &= \mathbb{E}\left[\sum_{t=1}^{T} R_t \mathbb{I}\{\tau_1 \leq T\}\right] + \mathbb{E}\left[\sum_{t=1}^{T} R_t \mathbb{I}\{\tau_1 > T\}\right] \\ &\overset{(a)}{\leq} T\mathbb{P}\left(\tau_1 \leq T\right) + \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} R_t \mathbb{I}\{\tau_1 > T\}\right]}_{(b)},\end{aligned}$$

where inequality (a) holds due to the fact that $R_t \leq 1$ and $\mathbb{E}\left[\mathbb{I}\{\tau_1 \leq T\}\right] = \mathbb{P}\left(\tau_1 \leq T\right)$.

In order to bound the term (b), we denote the event $\mathcal{U}$ as the algorithm being in the forced uniform exploration phase and let $\mathcal{E}_t := \{\exists \ell \in \mathcal{L} \text{ s.t. } |\mathbf{w}^1(\ell) - \hat{\mathbf{w}}_t(\ell)| \geq \sqrt{3\log t/(2n_{\ell,t})}\}$ be the event that $\hat{\mathbf{w}}_t(\ell)$ is not in the high-probability confidence interval around $\mathbf{w}^1(\ell)$, where $\mathbf{w}^1(\ell)$ is expected attraction of item ℓ in the first piecewise-stationary segment, $\hat{\mathbf{w}}_t(\ell)$ is the sample mean of item ℓ up to time slot t, and $n_{\ell,t}$ is the number of times that item ℓ is observed up to time slot t. Term (b) can be further decomposed as

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T} R_t \mathbb{I}\{\tau_1 > T\}\right] &= \mathbb{E}\left[\sum_{t=1}^{T} R_t \mathbb{I}\{\mathcal{U}\}\right] + \mathbb{E}\left[\sum_{t=1}^{T} R_t \mathbb{I}\{\tau_1 > T, \mathcal{E}_{t-1}, \overline{\mathcal{U}}\}\right] \\ &\quad + \mathbb{E}\left[\sum_{t=1}^{T} R_t \mathbb{I}\{\tau_1 > T, \overline{\mathcal{E}}_{t-1}, \overline{\mathcal{U}}\}\right] \\ &\overset{(c)}{\leq} Tp + \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} R_t \mathbb{I}\{\tau_1 > T, \mathcal{E}_{t-1}, \overline{\mathcal{U}}\}\right]}_{(d)} \\ &\quad + \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} R_t \mathbb{I}\{\tau_1 > T, \overline{\mathcal{E}}_{t-1}, \overline{\mathcal{U}}\}\right]}_{(e)},\end{aligned}$$

where inequality (c) is because of the fact that $R_t \leq 1$ and the uniform exploration probability is p. Term (d) can be bounded by applying the

Chernoff-Hoeffding inequality,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} R_t \mathbb{I}\{\tau_1 > T, \mathcal{E}_{t-1}, \overline{\mathcal{U}}\}\right] \\
&\leq \sum_{\ell=1}^{L}\sum_{t=1}^{T}\sum_{n_\ell=1}^{t} \mathbb{P}\left(|\mathbf{w}^1(\ell) - \hat{\mathbf{w}}_t(\ell)| \geq \sqrt{3\log t/(2n_\ell)}\right) \\
&\leq 2\sum_{\ell=1}^{L}\sum_{t=1}^{T}\sum_{n_\ell=1}^{t} e^{-3\log t} \leq \frac{\pi^2 L}{3}.
\end{aligned}$$

Furthermore, term (e) can be bounded as follows,

$$\mathbb{E}\left[\sum_{t=1}^{T} R_t \mathbb{I}\{\tau_1 > T, \overline{\mathcal{E}}_{t-1}, \overline{\mathcal{U}}\}\right] \overset{(f)}{\leq} pT + \sum_{\ell=K+1}^{L} \frac{12}{\Delta^1_{s_1(\ell),s_1(K)}} \log T,$$

where the inequality (f) follows the proof of Theorem 2 in [6]. By summing all terms, we prove the result. □

Then we bound the false alarm probability $\mathbb{P}(\tau_1 \leq T)$ in Lemma 4.1 under previously mentioned stationary scenario.

Lemma 4.2. *Consider the stationary scenario, with confidence level* $\delta \in (0,1)$ *for the Bernoulli GLRT, and we have that*

$$\mathbb{P}(\tau_1 \leq T) \leq L\delta.$$

Proof of Lemma 4.2. Define $\tau_{\ell,1}$ as the first change-point detection time of the ℓth item. Then, $\tau_1 = \min_{\ell\in\mathcal{L}} \tau_{\ell,1}$. Since the global restart is adopted by applying the union bound we have that

$$\mathbb{P}(\tau_1 \leq T) \leq \sum_{\ell=1}^{L} \mathbb{P}(\tau_{\ell,1} \leq T).$$

Recall the GLR statistic defined in (4.4), and plug it into $\mathbb{P}(\tau_{\ell,1} \leq T)$, we have that

$$\begin{aligned}
\mathbb{P}(\tau_{\ell,1} \leq \tau) \leq \mathbb{P}[&\exists (s,n) \in \mathbb{N}^2, n \leq n_\ell, s < n : \\
&s\mathrm{KL}\left(\hat{\mu}^1_{\ell,1:s}, \hat{\mu}^1_{\ell,1:n}\right) + (n-s)\mathrm{KL}\left(\hat{\mu}^1_{\ell,s+1:n}, \hat{\mu}^1_{\ell,1:n}\right) > \beta(n,\delta)]
\end{aligned}$$

$$
\begin{aligned}
&\leq \mathbb{P}[\exists (s,n) \in \mathbb{N}^2, n \leq T, s < n : \\
&\qquad s\mathrm{KL}\left(\hat{\mu}^1_{\ell,1:s}, \hat{\mu}^1_{\ell,1:n}\right) + (n-s)\mathrm{KL}\left(\hat{\mu}^1_{\ell,s+1:n}, \hat{\mu}^1_{\ell,1:n}\right) > \beta(n,\delta)] \\
&\overset{(a)}{\leq} \mathbb{P}[\exists (s,n) \in \mathbb{N}^2, n \leq T, s < n : \\
&\qquad s\mathrm{KL}\left(\hat{\mu}^1_{\ell,1:s}, \mathbf{w}^1(\ell)\right) + (n-s)\mathrm{KL}\left(\hat{\mu}^1_{\ell,s+1:n}, \mathbf{w}^1(\ell)\right) > \beta(n,\delta)] \\
&\overset{(b)}{\leq} \sum_{s=1}^{T} \mathbb{P}[\exists s < n : s\mathrm{KL}\left(\hat{\mu}^1_{\ell,1:s}, \mathbf{w}^1(\ell)\right) \\
&\qquad + (n-s)\mathrm{KL}\left(\hat{\mu}^1_{\ell,s+1:n}, \mathbf{w}^1(\ell)\right) > \beta(n,\delta)] \\
&\leq \sum_{s=1}^{T} \mathbb{P}[\exists r \in \mathbb{N} : s\mathrm{KL}\left(\hat{\mu}^1_{\ell,s}, \mathbf{w}^1(\ell)\right) \\
&\qquad + r\mathrm{KL}\left(\hat{\mu}^1_{k,r}, \mathbf{w}^1(\ell)\right) > \beta(s+r,\delta)] \\
&\overset{(c)}{\leq} \sum_{s=1}^{T} \frac{\delta}{3s^{3/2}} \overset{(d)}{\leq} \sum_{s=1}^{\infty} \frac{\delta}{3s^{3/2}} \leq \delta,
\end{aligned}
$$

where $\hat{\mu}^1_{\ell,s:s'}$ is the mean of the rewards generated from the distribution f^1_ℓ with expected reward $\mathbf{w}^1(\ell)$ from time slot s to s'. Inequality (a) is because of the fact that

$$
\begin{aligned}
&s\mathrm{KL}\left(\hat{\mu}_{1:s}, \hat{\mu}_{1:n}\right) + (n-s)\mathrm{KL}\left(\hat{\mu}_{s+1:n}, \hat{\mu}_{1:n}\right) \\
&= \inf_{\lambda \in [0,1]} \left[s\mathrm{KL}\left(\hat{\mu}_{1:s}, \lambda\right) + (n-s)\mathrm{KL}\left(\hat{\mu}_{s+1:n}, \lambda\right)\right];
\end{aligned}
$$

inequality (b) is because of the union bound; inequality (c) is because of the Lemma 10 in [62]; and inequality (d) holds due to the Riemann zeta function $\zeta(x)$ and when $x = 3/2$, $\zeta(3/2) < 2.7$. Thus, we conclude by $\mathbb{P}\left(\tau_1 \leq T\right) \leq L\delta$. □

Next, we define the event $\mathcal{C}^{(i)}$ that all the change-points up to ith have been detected quickly and correctly:

$$
\mathcal{C}^{(i)} = \left\{\forall j \leq i, \tau_j \in \{\nu_j + 1, \cdots, \nu_j + d_j\}\right\}. \tag{4.11}
$$

Lemma 4.3 below shows $\mathcal{C}^{(i)}$ happens with high probability.

Lemma 4.3. *(Lemma 12 in [62]) When $\mathcal{C}^{(i-1)}$ holds,* GLRT *with confidence level δ is capable of detecting the change point ν_i correctly and quickly with* *high probability, that is,*

$$\mathbb{P}\left[\tau_i \leq \nu_i | \mathcal{C}^{(i-1)}\right] \leq L\delta, \ \textit{and}\ \mathbb{P}\left[\tau_i \geq \nu_i + d_i | \mathcal{C}^{(i-1)}\right] \leq \delta,$$

where τ_i is the detection time of ith change-point.

In the next lemma, we bound the expected detection delay with the good event $\mathcal{C}^{(i)}$ holds.

Lemma 4.4. *The expected delay given $\mathcal{C}^{(i)}$ is:*

$$\mathbb{E}\left[\tau_i - \nu_i | \mathcal{C}^{(i)}\right] \leq d_i.$$

Proof. By the definition of $\mathcal{C}^{(i)}$, the conditional expected delay is obviously upper bounded by d_i. □

4.5.2 Proof of Theorem 4.1

Proof. Define good events $E_i = \{\tau_i > \nu_i\}$ and $D_i = \{\tau_i \leq \nu_i + d_i\}, \forall 1 \leq i \leq N-1$. Recall the definition of the good event $\mathcal{C}^{(i)}$ that all the change-points up to ith one have been detected correctly and quickly in (4.11), and we can find that $\mathcal{C}^{(i)} = E_1 \cap D_1 \cap \cdots \cap E_i \cap D_i$. Again, we denote $R_t := R\left(\mathcal{A}_t, \mathbf{w}_t, \mathbf{Z}_t\right)$ as the regret of the learning algorithm at time slot t. By first decomposing the expected cumulative regret with respect to the event E_1, we have that

$$\begin{aligned}
\mathcal{R}(T) &= \mathbb{E}\left[\sum_{t=1}^{T} R_t \mathbb{I}\{E_1\}\right] + \mathbb{E}\left[\sum_{t=1}^{T} R_t \mathbb{I}\{\overline{E_1}\}\right] \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T} R_t \mathbb{I}\{E_1\}\right] + T\mathbb{P}(\overline{E}_1) \\
&\overset{(a)}{\leq} \mathbb{E}\left[\sum_{t=1}^{\nu_1} R_t \mathbb{I}\{E_1\}\right] + \mathbb{E}\left[\sum_{t=\nu_1+1}^{T} R_t\right] + TL\delta \\
&\overset{(b)}{\leq} \widetilde{C}_1 + \nu_1 p + \underbrace{\mathbb{E}\left[\sum_{t=\nu_1+1}^{T} R_t\right]}_{(c)} + TL\delta,
\end{aligned}$$

where the inequality (a) is because that $\mathbb{P}(\overline{E}_1)$ can be bounded using Lemma 4.2 and inequality (b) holds due to Lemma 4.1. To bound the term (c), by applying

plying the law of total expectation, we have that

$$\begin{aligned}\mathbb{E}\left[\sum_{t=\nu_1+1}^{T} R_t\right] &\leq \mathbb{E}\left[\sum_{t=\nu_1+1}^{T} R_t \mid \mathcal{C}^{(1)}\right] + T(1-\mathbb{P}(E_1 \cap D_1)) \\ &= \mathbb{E}\left[\sum_{t=\nu_1+1}^{T} R_t \mid \mathcal{C}^{(1)}\right] + T(\mathbb{P}(\overline{E}_1 \cup \overline{D}_1)) \\ &\leq \underbrace{\mathbb{E}\left[\sum_{t=\nu_1+1}^{T} R_t \mid E_1 \cap D_1\right]}_{(d)} + T(L+1)\delta,\end{aligned}$$

where $\mathbb{P}(\overline{E}_1 \cup \overline{D}_1)$ is acquired by applying the union bound on the Lemma 4.3. Then, we turn to the term (d), by further splitting the regret,

$$\begin{aligned}\mathbb{E}\left[\sum_{t=\nu_1+1}^{T} R_t \mid E_1 \cap D_1\right] &= \mathbb{E}\left[\sum_{t=\nu_1+1}^{T} R_t \mid \mathcal{C}^{(1)}\right] \\ &\leq \mathbb{E}\left[\sum_{t=\tau_1+1}^{T} R_t \mid \mathcal{C}^{(1)}\right] + \underbrace{\mathbb{E}\left[\sum_{t=\nu_1+1}^{\tau_1} R_t \mid \mathcal{C}^{(1)}\right]}_{(e)} \\ &\leq \mathbb{E}\left[\sum_{t=\nu_1+1}^{T} R_t \mid \mathcal{C}^{(1)}\right] + d_1,\end{aligned}$$

where term (e) is bounded by applying the Lemma 4.4 and the fact that $R_t \leq 1$. Thus,

$$\mathcal{R}(T) \leq \mathbb{E}\left[\sum_{t=\nu_1+1}^{T} R_t \mid \mathcal{C}^{(1)}\right] + \widetilde{C}_1 + \nu_1 p + d_1 + 3TL\delta.$$

Similarly,

$$\begin{aligned}\mathbb{E}\left[\sum_{t=\nu_1+1}^{T} R_t \mid \mathcal{C}^{(1)}\right] &\leq \mathbb{E}\left[\sum_{t=\nu_1+1}^{T} R_t \mathbb{I}\{E_2\} \mid \mathcal{C}^{(1)}\right] + T\mathbb{P}(\overline{E}_2 | \mathcal{C}^{(1)}) \\ &\leq \mathbb{E}\left[\sum_{t=\nu_1+1}^{\nu_2} R_t \mathbb{I}\{E_2\} \mid \mathcal{C}^{(1)}\right] + \mathbb{E}\left[\sum_{t=\nu_2+1}^{T} R_t \mid \mathcal{C}^{(1)}\right] + TL\delta\end{aligned}$$

$$\leq \widetilde{C}_2 + (\nu_2 - \nu_1)p + \underbrace{\mathbb{E}\left[\sum_{t=\nu_2+1}^{T} R_t \mid \mathcal{C}^{(1)}\right]}_{(f)} + TL\delta,$$

where $\mathbb{P}(\overline{E}_2|\mathcal{C}^{(1)})$ directly follows Lemma 4.3. To bound term (f),

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=\nu_2+1}^{T} R_t \mid \mathcal{C}^{(1)}\right] \\
&\leq \mathbb{E}\left[\sum_{t=\nu_2+1}^{T} R_t \mid E_2 \cap D_2 \cap \mathcal{C}^{(1)}\right] + T(1 - \mathbb{P}(E_2 \cap D_2|\mathcal{C}^{(1)})) \\
&= \mathbb{E}\left[\sum_{t=\nu_2+1}^{T} R_t \mid E_2 \cap D_2 \cap \mathcal{C}^{(1)}\right] + T\mathbb{P}(\overline{E}_2 \cup \overline{D}_2|\mathcal{C}^{(1)}) \\
&\leq \underbrace{\mathbb{E}\left[\sum_{t=\nu_2+1}^{T} R_t \mid \mathcal{C}^{(2)}\right]}_{(g)} + T(L+1)\delta,
\end{aligned}$$

where $\mathbb{P}(\overline{E}_2 \cup \overline{D}_2|\mathcal{C}^{(1)})$ is acquired by applying the union bound on Lemma 4.3. For term (g), we have

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=\nu_2+1}^{T} R_t \mid \mathcal{C}^{(2)}\right] &\leq \mathbb{E}\left[\sum_{t=\tau_2+1}^{T} R_t \mid \mathcal{C}^{(2)}\right] + \mathbb{E}\left[\sum_{t=\nu_2+1}^{\tau_2} R_t \mid \mathcal{C}^{(2)}\right] \\
&\leq \mathbb{E}\left[\sum_{t=\nu_2+1}^{T} R_t \mid \mathcal{C}^{(2)}\right] + d_2.
\end{aligned}$$

Wrapping up previous steps, we have that

$$\mathcal{R}(T) \leq \mathbb{E}\left[\sum_{t=\nu_2+1}^{T} R_t \mid \mathcal{C}^{(2)}\right] + \widetilde{C}_1 + \widetilde{C}_2 + \nu_2 p + d_1 + d_2 + 6TL\delta.$$

Recursively, the upper bound on the regret of `GLRT-CascadeUCB` is given by

$$\mathcal{R}(T) \leq \sum_{i=1}^{N} \widetilde{C}_i + Tp + \sum_{i=1}^{N-1} d_i + 3NTL\delta.$$

□

4.5.3 Proof of Corollary 4.1

Proof. By applying the upper bound on $\mathcal{G}(x)$ that $\mathcal{G}(x) \leq x + 4\log(1 + x + \sqrt{2x})$ if $x \geq 5$ to d_i, we have that

$$
\begin{aligned}
d_i &\leq \frac{4L}{p\left(\Delta_{\text{change}}^{\min}\right)^2 \beta(T,\delta)} + \frac{2L}{p} \\
&\overset{(a)}{\leq} \frac{4L}{p\left(\Delta_{\text{change}}^{\min}\right)^2} [\log\left(\frac{3T^{3/2}}{\delta}\right) + 8\log\left(1 + \frac{\log(\frac{3T^{3/2}}{\delta})}{2} + \sqrt{\log\left(\frac{3T^{3/2}}{\delta}\right)}\right) \\
&\quad + 6\log(1+\log T)] + \frac{2L}{p} \\
&\overset{(b)}{\leq} \frac{\frac{20L\log T + o(L\log T)}{\left(\Delta_{\text{change}}^{\min}\right)^2} + 2L}{p} \lesssim \frac{L\log T}{p\left(\Delta_{\text{change}}^{\min}\right)^2},
\end{aligned}
$$

where $(a)(b)$ hold when $\log(3T^{5/2}) \geq 10$ (equals to $T \geq 36$). By plugging d_i into Theorem 4.1, we have that,

$$
\mathcal{R}(T) \lesssim \frac{N(L-K)\log T}{\Delta_{\text{opt}}^{\min}} + Tp + \frac{NL\log T}{p\left(\Delta_{\text{change}}^{\min}\right)^2} + 3NL.
$$

Combining the above analysis we conclude the corollary. □

4.6 Proof of Theorem 4.2

Proof of Theorem 4.2. We start by defining the good event $\mathcal{H}_T$ that all the change-points have been detected correctly and quickly,

$$
\mathcal{H}_T := \{\forall i = 1, \ldots, N-1, \tau_i \in \{\nu_i + 1, \ldots, \nu_i + d_i\}\},
$$

And let $\mathcal{E}_{t,i} := \{\exists \ell \in \{s_i(1), \ldots, s_i(K)\} \text{ s.t. } \mathbf{w}^i(\ell) > \text{UCB}_{\text{KL},t}(\ell)\}$ be the event that the expected attraction of at least one optimal item is above the UCB index at time slot t and t is in ith piecewise-stationary segment, where $\text{UCB}_{\text{KL},t}(\ell)$ is the KL-UCB index of ℓ item computed at time slot t. The regret of `GLRT-CascadeKL-UCB` can be decomposed as

$$
\mathcal{R}(T) \leq \mathbb{E}\left[\sum_{t=1}^{T} R_t \mathbb{I}\{\mathcal{U}\}\right] + \mathbb{E}\left[\sum_{t=1}^{T} R_t \mathbb{I}\{\overline{\mathcal{U}}, \overline{\mathcal{H}}_T\}\right] + \mathbb{E}\left[\sum_{t=1}^{T} R_t \mathbb{I}\{\overline{\mathcal{U}}, \mathcal{H}_T\}\right]
$$

$$\leq pT + \underbrace{T\mathbb{P}(\overline{\mathcal{H}}_T)}_{(a)} + \sum_{i=1}^{N-1} d_i + \underbrace{\mathbb{E}\left[\sum_{t=1}^{\nu_1} R_t \mathbb{I}\{\overline{\mathcal{U}}, \mathcal{H}_T, \mathcal{E}_{t-1,1}\}\right]}_{(b)}$$
$$+ \sum_{i=1}^{N-1} \underbrace{\mathbb{E}\left[\sum_{t=\tau_i+1}^{\nu_{i+1}} R_t \mathbb{I}\{\overline{\mathcal{U}}, \mathcal{H}_T, \mathcal{E}_{t-1,i+1}\}\right]}_{(c)}$$
$$+ \underbrace{\mathbb{E}\left[\sum_{t=1}^{\nu_1} R_t \mathbb{I}\{\overline{\mathcal{U}}, \mathcal{H}_T, \overline{\mathcal{E}}_{t-1,1}\}\right]}_{(d)} + \sum_{i=1}^{N-1} \underbrace{\mathbb{E}\left[\sum_{t=\tau_i+1}^{\nu_{i+1}} R_t \mathbb{I}\{\overline{\mathcal{U}}, \mathcal{H}_T, \overline{\mathcal{E}}_{t-1,i+1}\}\right]}_{(e)}.$$

Bound Term (a): Recall the definition of $\mathcal{C}^{(i)}$ and applying the union bound,

$$\begin{aligned}
\mathbb{P}(\overline{\mathcal{H}}_T) &\leq \sum_{i=1}^{N-1} \mathbb{P}(\tau_i \notin \{\nu_i + 1, \ldots, \nu_i + d_i\} | \mathcal{C}^{(i-1)}) \\
&\leq \sum_{i=1}^{N-1} \mathbb{P}(\tau_i \leq \nu_i | \mathcal{C}^{(i-1)}) + \sum_{i=1}^{N-1} \mathbb{P}(\tau_i \geq \nu_i + d_i | \mathcal{C}^{(i-1)}) \\
&\leq (N-1)(L+1)\delta,
\end{aligned}$$

where the last inequality is due to Lemma 4.3.

Bound Terms (b) and (c): By plugging in the event $\mathcal{E}_{t,i}$, we have that

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=\tau_i+1}^{\nu_{i+1}} R_t \mathbb{I}\{\overline{\mathcal{U}}, \mathcal{H}_T, \mathcal{E}_{t-1,i+1}\}\right] \\
&\leq \sum_{\ell^*=1}^{K} \mathbb{E}\left[\mathbb{I}\{\mathcal{C}^{(i)}\} \sum_{t=\tau_i+1}^{\nu_{i+1}} \mathbb{I}\{n_{s_t(\ell^*),t} \mathrm{KL}(\hat{\mathbf{w}}_t(s_t(\ell^*)), \mathbf{w}_t(s_t(\ell^*))) \geq g(t-\tau_i)\}\right] \\
&\leq K\mathbb{E}\left[\sum_{t=\tau_i+1}^{\nu_{i+1}} \mathbb{I}\{n_{s_t(\ell^*),t} \mathrm{KL}(\hat{\mathbf{w}}_t(s_t(\ell^*)), \mathbf{w}_t(s_t(\ell^*))) \geq g(t-\tau_i)\} | \mathcal{C}^{(i)}\right] \\
&= K\mathbb{E}\left[\sum_{t=\tau_i+1}^{\nu_{i+1}} \mathbb{I}\{n_{s_t(\ell^*),t} \mathrm{KL}(\hat{\mathbf{w}}_t(s_t(\ell^*)), \mathbf{w}^i(s_t(\ell^*))) \geq g(t-\tau_i)\} | \mathcal{C}^{(i)}\right] \\
&\leq K \sum_{t'=1}^{\nu_{i+1}-\tau_i} \mathbb{P}(\exists s \leq t' : s\mathrm{KL}(\hat{\mu}_s, \mathbf{w}^i(s_t(\ell^*))) \geq g(t'))
\end{aligned}$$

$$\leq K \sum_{t=1}^{T} \frac{1}{t \log t} \leq K \log \log T,$$

where the first inequality is due to $\mathcal{H}_T \in \mathcal{C}^{(i)}$; $\hat{\mathbf{w}}(\ell)$ is the mean of the rewards of item ℓ after the most recent detection time τ and up to time slot t; and the last inequality follows directly from Lemma 2 in [212]. Note that (b) can be upper bounded similar to the procedures of bounding (c).

Bound Terms (d) and (e): Here, according to the proof of Theorem 3 in [6], (d) and (e) can be bounded as

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{\nu_1} R_t \mathbb{I}\{\overline{\mathcal{U}}, \mathcal{H}_T, \overline{\mathcal{E}}_{t-1,1}\}\right] \text{ or } \mathbb{E}\left[\sum_{t=\tau_i+1}^{\nu_{i+1}} R_t \mathbb{I}\{\overline{\mathcal{U}}, \mathcal{H}_T, \overline{\mathcal{E}}_{t-1,i+1}\}\right] \\
&\leq \sum_{\ell=K+1}^{L} \frac{(1+\epsilon)\Delta^{i+1}_{s_{i+1}(\ell),s_{i+1}(K)}(1+\log(1/\Delta^{i+1}_{s_{i+1}(\ell),s_{i+1}(K)})}{\mathrm{KL}(\mathbf{w}^{i+1}(s_{i+1}(\ell)), \mathbf{w}^{i+1}(s_{i+1}(K)))}(\log T + 3\log\log T) \\
&\qquad + \frac{C_2(\epsilon)}{d_i^{\beta(\epsilon)}},
\end{aligned}$$

where $C_2(\epsilon)$ and $\beta(\epsilon)$ follow the same definition in [6]. Denote $\widetilde{D}_i$ as

$$\begin{aligned}
\widetilde{D}_i = &\sum_{\ell=K+1}^{L} \frac{(1+\epsilon)\Delta^{i+1}_{s_{i+1}(\ell),s_{i+1}(K)}(1+\log(1/\Delta^{i+1}_{s_{i+1}(\ell),s_{i+1}(K)}))}{\mathrm{KL}(\mathbf{w}^{i+1}(s_{i+1}(\ell)), \mathbf{w}^{i+1}(s_{i+1}(K)))} \\
&\times (\log T + 3\log\log T) + \frac{C_2(\epsilon)}{d_i^{\beta(\epsilon)}}.
\end{aligned} \tag{4.12}$$

Summing up all terms, and we have that

$$\mathcal{R}(T) \leq T(N-1)(L+1)\delta + Tp + \sum_{i=1}^{N-1} d_i + KN\log\log T + \sum_{i=0}^{N-1} \widetilde{D}_i.$$

□

4.7 Proof of Theorem 4.3

Proof of Theorem 4.3. The first step in deriving the minimax lower bound is to construct a randomized 'hard instance' as follows. Partition the time horizon T into N blocks and name them $B_1, \ldots, B_N$, where the lengths of first

$N-1$ blocks are $\lceil T/N \rceil$ and the length of the last block is $T-(N-1)\lceil T/N \rceil$. In each segment, $L-1$ items follow Bernoulli distribution with probability $1/2$ and only one item follows Bernoulli distribution with probability $1/2+\epsilon$, where ϵ is a small positive number. Let $\ell_i^* = \arg\max_{\ell \in \mathcal{L}} \mathbf{w}^i(\ell)$, i.e, the item with largest click probability during B_i. The distributions of the ℓ_i^*'s are defined as follows:

- $\ell_1^* \sim \text{Uniform}(\{1, \ldots, L\})$.
- for $i \geq 2$, $\ell_i^* \sim \text{Uniform}(L \setminus \ell_{i-1}^*)$.

Note that for this randomized instance, the regret for any policy π is

$$\mathcal{R}^\pi(T) = \epsilon(1/2)^{K-1}\mathbb{E}^\pi[\sum_{i=1}^{N}\sum_{t\in B_i}\mathbb{I}\{\ell_i^* \notin \mathcal{A}_t\}].$$

The expectation is taken with respect to the policy π and this randomized instance. From the above decomposition, we see that to lower bound the regret for any policy π, it suffices to upper bound $\mathbb{E}^\pi[\sum_{i=1}^{N}\sum_{t\in B_i}\mathbb{I}\{\ell_i^* \in \mathcal{A}_t\}]$, the expectation of total number of recommendations to the item with largest click probability. Before we lower bound this quantity, we need some additional notation. Let P_i^ℓ be the joint distribution of $\{\mathcal{A}_t, F_t\}_{t\in B_i}$ given the policy π and the ℓth item being the item with largest click probability, P_i^0 be the joint distribution of $\{\mathcal{A}_t, F_t\}_{t\in B_i}$ given the policy π and every item following the Bernoulli distribution with probability $1/2$. Furthermore, let $\mathbb{E}_i^\ell[\cdot]$ and $\mathbb{E}_i^0[\cdot]$ as their respective expectations. Let N_i^ℓ be the total numbers of appearances of item ℓ in the recommendation list during B_i. In order to lower bound the target expectation, we need the following lemma.

Lemma 4.5. *For any segment B_i and any $\ell \in \mathcal{L}$, we have*

$$\mathbb{E}_i^\ell[N_i^\ell] \leq \mathbb{E}_i^0[N_i^\ell] + \frac{|B_i|}{2}\sqrt{\mathbb{E}_i^0[N_i^\ell]\log(\frac{1}{1-4\epsilon^2})}.$$

Proof of Lemma 4.5. The proof is similar to Lemma A.1 in [217]. The key difference is we apply the data processing inequality for KL divergence to upper bound the discrepancy of the partial feedback F_t's under different distributions.

$$\begin{aligned}
\mathbb{E}_i^{\ell}[N_i^l] - \mathbb{E}_i^0[N_i^{\ell}] &\overset{(a)}{\leq} \frac{|B_i|}{2} \left\| P_i^{\ell} - P_i^0 \right\|_1 \\
&\overset{(b)}{\leq} \frac{|B_i|}{2} \sqrt{2 D_{\mathrm{KL}}(P_i^0 || P_i^{\ell})} \\
&= \frac{|B_i|}{2} \sqrt{2 \sum_{t \in B_i} D_{\mathrm{KL}}(P_i^0(F_t|\mathcal{A}_t) || P_i^{\ell}(F_t|\mathcal{A}_t))} \\
&\overset{(c)}{\leq} \frac{|B_i|}{2} \sqrt{2 \sum_{t \in B_i} D_{\mathrm{KL}}(P_i^0(\mathbf{Z}_t|\mathcal{A}_t) || P_i^{\ell}(\mathbf{Z}_t|\mathcal{A}_t))} \\
&= \frac{|B_i|}{2} \sqrt{\mathbb{E}_i^0[N_i^{\ell}] \log(\frac{1}{1-4\epsilon^2})},
\end{aligned}$$

where $D_{\mathrm{KL}}(\cdot)$ is the KL divergence, (a) is due to the boundedness of N_i^l, (b) is due to Pinsker's inequality, (c) is due to data processing inequality for KL divergence. □

Apply Lemma 4.5 for B_i and sum over all items to get

$$\begin{aligned}
\sum_{\ell \in \mathcal{L}} \mathbb{E}_i^{\ell}[N_i^{\ell}] &\leq \sum_{\ell \in \mathcal{L}} \mathbb{E}_i^0[N_i^{\ell}] + \sum_{\ell \in \mathcal{L}} \frac{|B_i|}{2} \sqrt{\mathbb{E}_i^0[N_i^{\ell}] \log(\frac{1}{1-4\epsilon^2})} \\
&\leq |B_i| + \frac{|B_i|}{2} \sqrt{|B_i| L \log(\frac{1}{1-4\epsilon^2})}, \qquad (4.13)
\end{aligned}$$

where the last inequality is due to $\sum_{\ell \in \mathcal{L}} \mathbb{E}_i^0[N_i^{\ell}] = |B_i|$ and Jensen's inequality. Then we are able to lower bound the regret for any policy π.

$$\begin{aligned}
\mathcal{R}^{\pi}(T) &= \epsilon(1/2)^{K-1} \left(T - \mathbb{E}^{\pi}[\sum_{i=1}^{N} \sum_{t \in B_i} \mathbb{I}\{\ell_i^* \in \mathcal{A}_t\}] \right) \\
&\overset{(a)}{\geq} (1/2)^{K-1} \epsilon \left(T - \tfrac{1}{L-1} (\sum_{i=1}^{N} |B_i| + \frac{|B_i|}{2} \sqrt{L |B_i| \log \tfrac{1}{1-4\epsilon^2}} \right) \\
&= (1/2)^{K-1} \left(\epsilon T - \tfrac{\epsilon T}{L-1} - \tfrac{\epsilon T}{2(L-1)} \sqrt{\frac{LT}{N} \log \tfrac{1}{1-4\epsilon^2}} \right) \\
&\overset{(b)}{\geq} (1/2)^{K-1} \left(\tfrac{\epsilon T}{2} - \tfrac{\epsilon^2 T}{K-1} \sqrt{\tfrac{LT}{N} \log \tfrac{4}{3}} \right).
\end{aligned}$$

where (a) is due to inequality (4.13), and (b) holds by $L \geq 3$, $4\epsilon^2 \leq \frac{1}{4}$ and $\log \frac{1}{1-x} \leq 4 \log(\frac{4}{3}) x$ for all $x \in [0, \frac{1}{4}]$. Finally, setting $\epsilon = \frac{L-1}{4\sqrt{TL \log(\frac{4}{3})}}$ finishes the proof. □

CHAPTER 5

ADVERSARIAL LINEAR CONTEXTUAL BANDITS WITH GRAPH-STRUCTURED SIDE OBSERVATIONS

5.1 Problem Formulation

Notation. In Chapter 5, we use $\|x\|_2$ to denote the Euclidean norm of vector x; $\langle x, y\rangle$ stands for the inner product of x and y. We also define $\mathbb{E}_t\left[\cdot\right] = \mathbb{E}\left[\cdot\middle|\mathcal{F}_{t-1}\right]$ as the expectation given the filtration $\mathcal{F}_{t-1}$.

We consider an adversarial linear contextual bandit problem with graph-structured side observations between an agent with a finite action set $V := \{1, \ldots, L\}$ and its adversary. At each time step $t = 1, 2, \ldots, T$, the interaction steps between the agent and its adversary are repeated, which are described as follows. At the beginning of time step t, the feedback graph $G_t(V, \mathcal{E}_t)$ and a loss vector $\theta_{i,t} \in \mathbb{R}^d$ for each action $i \in V$ are chosen by the adversary arbitrarily, where G_t can be directed or undirected, V is the node set (the same as the action set V), and $\mathcal{E}_t$ is the edge set. Note that G_t and $\theta_{i,t}$ are *not* disclosed to the agent at this time. After observing a context $X_t \in \mathbb{R}^d$, the agent chooses an action $I_t \in V$ to play based on X_t, the previous interaction history, and possibly some randomness in the policy, and incurs the loss $\ell_t(X_t, I_t) = \langle X_t, \theta_{I_t,t}\rangle$. Unlike recently proposed adversarial linear contextual bandits [81], where only the played action I_t discloses its loss $\ell_t(X_t, I_t)$, here we assume all losses in a subset $S_{I_t,t} \subseteq V$ are disclosed after I_t is played, where S_{I_t} contains I_t and its neighboring nodes in the feedback graph G_t. More formally, we have that $S_{i,t} := \{j \in V \big| i \xrightarrow{t} j \in \mathcal{E}_t \text{ or } j = i\}$, where $i \xrightarrow{t} j$ indicates an edge from node i to node j in a directed graph or an edge between i and j in an undirected graph at time t. These observations except for that of action I_t are called *side observations* in graphical bandits [71]. In addition, an *oracle* provides extra observations for all $i \in S_{I_t}$ (see Assumption 5.2 for details). Before proceeding to time step $t + 1$, the adversary discloses G_t to the agent.

Remark 5.1. The way the adversary discloses G_t in this chapter is called the **uninformed** setting, where G_t is disclosed **after** the agent's decision making. Contrarily, a simpler setting from the agent's perspective is called the **informed** setting [72], where G_t is disclosed **before** the agent's decision making. The uninformed setting is the minimum requirement for our problem to capture the benefits of side observations [218, Theorem 1].

Furthermore, we have the following assumptions for above interaction steps.

Assumption 5.1 (i.i.d. contexts)**.** *The context $X_t \in \mathbb{R}^d$ is drawn from a distribution $\mathcal{D}$ independently from the choice of loss vectors and other contexts, where $\mathcal{D}$ is known by the agent in advance .*

Assumption 5.2 (extra observation oracle)**.** *Assume at each time step t, there exists an **oracle** that draws a context $\tilde{X}_t \in \mathbb{R}^d$ from $\mathcal{D}$ independently from the choice of loss vectors and other contexts, and discloses $\tilde{X}_t$ together with the losses $\tilde{l}_t(\tilde{X}_t, i) = \left\langle \tilde{X}_t, \theta_{i,t} \right\rangle$ for all $i \in S_{I_t,t}$ to the agent.*

Assumption 5.3 (nonoblivious adversary)**.** *The adversary can be **nonoblivious**, who is allowed to choose G_t and $\theta_{i,t}, \forall i \in V$ at time t according to arbitrary functions of the interaction history $\mathcal{F}_{t-1}$ before time step t. Here, $\mathcal{F}_t := \sigma(X_s, \tilde{X}_s, I_s, G_s, \{\ell_s(X_s, i)\}_{i \in S_s}, \{\tilde{\ell}_s(\tilde{X}_s, i)\}_{i \in S_s}, \forall s \leq t)$ is the filtration capturing the interaction history up to time step t.*

Remark 5.2. Assumption 5.1 is standard in the literature of adversarial contextual bandits [81, 219, 220, 221]. In fact, it has been shown that if both the contexts and loss vectors are chosen by the adversary, no algorithm can achieve a sublinear regret [81, 220]. The oracle in Assumption 5.2 is mainly adopted from the proof perspective, and its role will be clear in the analysis. In real-world applications, this oracle can be realized. Consider the viral marketing problem for an example. After the user and her/his followers complete the questionnaire and get the offers, they will probably purchase the products and leave online reviews after they experience those products. Then, the extra observations can be provided by those reviews. Assumption 5.3 indicates $\theta_{t,i}$ is a random vector with $\mathbb{E}_t\left[\theta_{i,t}\right] = \theta_{i,t}$, and a similar result holds for G_t. Note that a bandit problem with a nonoblivious adversary is harder than that with an oblivious adversary [222, 223] that chooses all loss vectors and feedback graphs before the start of the interactions.

The goal of the agent is to find a policy that minimizes its *expected cumulative loss*. Equivalently, we can adopt the *expected cumulative (pseudo) regret*, defined as the maximum gap between the expected cumulative loss incurred by the agent and that of a properly chosen policy set Π,

$$\begin{aligned}\mathcal{R}_T &= \max_{\pi_T \in \Pi} \mathbb{E}\left[\sum_{t=1}^{T} \left\langle X_t, \theta_{I_t,t} - \theta_{\pi_T(X_t),t}\right\rangle\right] \\ &= \max_{\pi_T \in \Pi} \mathbb{E}\left[\sum_{t=1}^{T} \sum_{i \in V} \left(\pi_t^a(i|X_t) - \pi_T(i|X_t)\right) \langle X_t, \theta_{i,t}\rangle\right],\end{aligned}$$

where the expectation is taken over the randomness of the agent's policy and the contexts. It is widely acknowledged that competing with a policy that uniformly chooses the best action in each time step t while incurring an $o(T)$ regret is hopeless in the adversarial setting [222, 223]. Thus, we adopt the fixed policy set Π proposed for adversarial linear contextual bandits [81],

$$\Pi := \left\{\pi_T \middle| \text{all policies } \pi_T : \mathbb{R}^d \mapsto V\right\}, \tag{5.1}$$

where the decision given by $\pi_T \in \Pi$ only depends the current received context X_t. The best policy $\pi_T^* \in \Pi$ is the one that satisfies the following condition

$$\pi_T^*(i|x) = \mathbb{I}\left\{i = \arg\min_{j \in V} \sum_{t=1}^{T} \langle x, \mathbb{E}[\theta_{j,t}]\rangle\right\}, \forall x \in \mathbb{R}^d,$$

which can be derived from the regret definition as shown in [81].

Before presenting our algorithms, we will further introduce several common assumptions and definitions in linear contextual bandits and graphical bandits. We assume the context distribution $\mathcal{D}$ is supported on a bounded set with each $x \sim \mathcal{D}$ satisfying $\|x\|_2 \leq \sigma$ for some positive σ. Furthermore, we assume the covariance $\Sigma = \mathbb{E}[X_t X_t^\top]$ of $\mathcal{D}$ to be positive definite with its smallest eigenvalue being $\lambda_{\min} > 0$. As for the loss vector $\theta_{i,t}$, we assume that $\|\theta_{i,t}\|_2 \leq L$ for some positive L for all i, t. Additionally, the loss $\ell_t(x,t)$ is bounded in $[-1,1]$: $|\ell_t(x,i)| \leq 1$ for all $x \sim \mathcal{D}$, i, and t. We have the following graph-theoretic definition from [72, 74, 78].

Definition 5.1 (Independence number)**.** *The cardinality of the maximum independent set of a graph G_t is defined as the* ***independence number*** *and denoted by $\alpha(G_t)$, where an independence set of $G_t = (V_t, \mathcal{E}_t)$ is any subset*

$V_t' \in V_t$ such that no two nodes $i, j \in V_t'$ are connected by an edge in $\mathcal{E}_t$. Note that $\alpha(G_t) \leq L$ in general. Without ambiguity, we use $\alpha(G) := \frac{1}{T}\sum_{t=1}^{T}\alpha(G_t)$ to denote the average independence number *of the feedback graphs $\{G_t\}_{t=1}^{T}$ in remainder of Chapter 5.*

5.2 The `EXP3-LGC-U` Algorithm

Algorithm 5.1: `EXP3-LGC-U`

```
Require: : Learning rate $\eta > 0$, uniform exploration rate $\gamma \in (0, 1)$, covariance
    $\Sigma$, and action set $V$
for $t = 1, \ldots, T$ do
    Feedback graph $G_t$ and loss vectors $\{\theta_{i,t}\}_{i \in V}$ are generated but not
    disclosed
    Observe $X_t \sim \mathcal{D}$, and for all $i \in V$, set
        $$w_t(X_t, i) = \exp\left(-\eta \sum_{s=1}^{t-1} \left\langle X_t, \hat{\theta}_{i,s} \right\rangle\right) \qquad (5.2)$$
    Play action $I_t$ drawn according to distribution
    $\pi_t^a(X_t) := (\pi_t^a(1 \mid X_t), \ldots, \pi_t^a(L \mid X_t))$, where
        $$\pi_t^a(i \mid X_t) = (1-\gamma)\frac{w_t(X_t, i)}{\sum_{j \in V} w_t(X_t, j)} + \frac{\gamma}{L} \qquad (5.3)$$
    Observe pairs $(i, \ell_t(X_t, i))$ for all $i \in S_{I_t,t}$, and disclose feedback graph $G_t$
    Extra observation oracle: observe $\tilde{X}_t \sim \mathcal{D}$ and pairs $(i, \tilde{\ell}_t(\tilde{X}_t, i))$ for all
    $i \in S_{I_t,t}$
    For each $i \in V$, estimate the loss vector $\theta_{i,t}$ as
        $$\hat{\theta}_{i,t} = \frac{\mathbb{I}\{i \in S_{I_t,t}\}}{q_t(i \mid X_t)} \Sigma^{-1} \tilde{X}_t \tilde{\ell}_t(\tilde{X}_t, i), \qquad (5.4)$$
    where $q_t(i \mid X_t) = \pi_t^a(i \mid X_t) + \sum_{j: j \xrightarrow{t} i} \pi_t^a(j \mid X_t)$
end for
```

In this section, we introduce our first simple yet efficient algorithm, `EXP3-LGC-U`, for both directed and undirected feedback graphs, which is the abbreviation for "**EXP3** for **L**inear **G**raphical **C**ontextual bandits with **U**niform exploration". Detailed steps of `EXP3-LGC-U` are presented in Algorithm 5.1. The upper bounds for the regret of `EXP3-LGC-U` are developed in Section 5.2.1.

We further discuss our theoretical findings on `EXP3-LGC-U` in Section 5.2.2. The proofs for the Claims, Theorems, and Corollaries in this section are deferred to Section 5.4.

The core of our algorithm, similar to many other algorithms for adversarial bandits, is designing an appropriate estimator of each loss vector and using those estimators to define a proper policy. Following the `EXP3`-based algorithms, we apply an exponentially weighted method and play an action i with probability proportional to $\exp(-\eta \sum_{s=1}^{t-1} \langle X_t, \hat{\theta}_{i,s} \rangle)$ (see (5.2)) at time step t, where η is the learning rate. More precisely, a uniform exploration γ is needed for the probability distribution of drawing action (see (5.3)). The uniform exploration is to control the variance of the loss vector estimators, a key step in our analysis. At this point, the key remaining question is how to design a reasonable estimator for each loss vector $\theta_{i,t}$. The answer can be found in (5.4), which takes advantage of both the original observations and the extra observations from the oracle. Similar to `EXP3-SET`, our algorithm uses importance sampling to construct the loss vector estimator $\hat{\theta}_{i,t}$ with controlled variance. The term $q_t(i|X_t)$ in the denominator in (5.4) indicates the probability of observing the loss of action i at time t, which is simply the sum of all $\pi_t^a(j|X_t)$ for all j that is connected to i at time t. The reason we use $\tilde{\ell}(\tilde{X}_t, i)$ and $\tilde{X}_t$ instead of $\ell(\tilde{X}_t, i)$ and X_t in constructing loss vector estimator $\hat{\theta}_{i,t}$ can be partly interpreted in the following two claims.

Claim 5.1. *The estimator $\hat{\theta}_{i,t}$ of the loss vector $\theta_{i,t}$ in* (5.4) *is an unbiased estimator given the interaction history $\mathcal{F}_{t-1}$ and X_t, for each $i \in V$ and t, i.e.,* $\mathbb{E}_t \left[\hat{\theta}_{i,t} \middle| X_t \right] = \theta_{i,t}$.

It is straightforward to show that the estimator $\hat{\theta}_{i,t}$ in (5.4) is unbiased w.r.t. $\mathbb{E}_t [\cdot]$ and $\mathbb{E} [\cdot]$ by applying the law of total expectation. However, if we use X_t and $\ell(X_t, i)$ to construct $\hat{\theta}_{i,t}$ in (5.4), it will only be unbiased w.r.t. $\mathbb{E}_t [\cdot]$ and $\mathbb{E} [\cdot]$, but not $\mathbb{E}_t [\cdot| X_t]$. This observation turns out to be essential in our analysis, which leads to the following immediate result of Claim 5.1.

Claim 5.2. *Let $\pi_T : \mathbb{R}^d \mapsto V$ be any policy in Π and $\hat{\theta}_{i,t}$ follows* (5.4).

Suppose π_t^a is determined by $\mathcal{F}_{t-1}$ and X_t, we have

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i\in V}\left(\pi_t^a(i|X_t)-\pi_T(i|X_t)\right)\langle X_t,\theta_{i,t}\rangle\right]\\
&=\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i\in V}\left(\pi_t^a(i|X_t)-\pi_T(i|X_t)\right)\left\langle X_t,\hat{\theta}_{i,t}\right\rangle\right]. \qquad (5.5)
\end{aligned}
$$

Remark 5.3. The advantages and properties of Claim 5.2 are summarized as follows: i) By applying the policy produced by `EXP3-LGC-U` and the best policy in the fixed policy set Π in (5.1), the term in the right hand side of (5.5) is exactly the regret $\mathcal{R}_T$ of `EXP3-LGC-U`. Given this property, the known loss vector estimate $\hat{\theta}_{i,t}$, instead of the unknown true loss vector $\theta_{i,t}$, can be applied directly to our analysis of the regret. ii) Claim 5.2 is not confined to `EXP3-LGC-U` and can be applied to other loss vector estimators that adopt different construction methods and any other benchmark policy, as long as Claim 5.1 is satisfied. iii) Based on Claim 5.2, some techniques in proving classical `EXP3` can be utilized in our analysis of the regret.

Remark 5.4. Claim 5.2 exhibits several differences between adversarial contextual bandits and classical adversarial MAB. First, the benchmark policy $\pi_T(\cdot|X_t)$ depends on the contexts in adversarial contextual bandits, while the benchmark policy is the best fixed action in hindsight in classical adversarial MAB. Second, consider the regret definition of classical adversarial MAB, $\mathcal{R}_T^{\text{MAB}} = \max_{j\in V}\mathbb{E}\left[\sum_{t=1}^{T}(\sum_{i\in V}\pi_t^{a,\text{MAB}}(i)\ell_{i,t})-\ell_{j,t}\right]$, where $\pi_t^{a,\text{MAB}}(i)$ is the policy produced by an `EXP3`-based algorithm and $\ell_{i,t}$ is the loss for action i at time step t. Since no context exists here, it is natural to design an estimator $\hat{\ell}_{i,t}$ of $\ell_{i,t}$ that is unbiased w.r.t. $\mathbb{E}_t[\cdot]$, and a similar result as Claim 5.2 can be proven. However, with the contexts, if the loss vector estimators are only unbiased w.r.t. $\mathbb{E}_t[\cdot]$ rather than $\mathbb{E}_t[\cdot|\,X_t]$, Claim 5.2 will not hold as shown in the proof of Claim 5.2 in Section 5.4.2.

Remarks 5.3 and 5.4 explain the need of adopting the extra observation oracle in `EXP3-LGC-U` and the way the loss vector estimator $\hat{\theta}_{i,t}$ is constructed.

5.2.1 Regret Analysis for `EXP3-LGC-U`

Our main theoretical justification for the performance of `EXP3-LGC-U` summarized in Theorem 5.1.

Theorem 5.1. *For any positive $\eta \in (0,1)$, choosing $\gamma = \eta L\sigma^2/\lambda_{min}$, the expected cumulative regret of* `EXP3-LGC-U` *satisfies:*

$$\mathcal{R}_t \leq \frac{\log L}{\eta} + \frac{2\eta L\sigma^2}{\lambda_{min}}T + \eta d \sum_{t=1}^{T} \mathbb{E}\left[Q_t\right],$$

where $Q_t = \alpha(G_t)$ if G_t is undirected, and $Q_t = 4\alpha(G_t)\log(4L^2/(\alpha(G_t)\gamma))$ if G_t is directed.

The proof of Theorem 5.1 is mainly based on the following Lemma 5.1, which is established on Claim 5.2.

Lemma 5.1. *Supposing $\left|\eta\left\langle X_t, \hat{\theta}_{i,t}\right\rangle\right| \leq 1$, the expected cumulative regret of* `EXP3-LGC-U` *satisfies*

$$\mathcal{R}_T \leq \frac{\log L}{\eta} + 2\gamma T + \eta\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i\in V}\pi_t^a(i|X_t)\left\langle X_t, \hat{\theta}_{i,t}\right\rangle^2\right]. \tag{5.6}$$

The proof of Lemma 5.1 is detailed in Section 5.4.3. The last term in the right side of (5.6) can be further bounded using graph-theoretic results in [74, Lemma 10] and [73, Lemma 5], which are restated in Section 5.4.

Remark 5.5. According to (5.13) in the proof of Theorem 5.1 in Section 5.4, if the extra observation oracle is not adopted, we will have a higher-order term $\mathbb{E}\left[X_t^\top\Sigma^{-1}X_tX_t^\top\Sigma^{-1}X_t\right]$. In general, it is hard to specify the relationship between this term and the dimension of contexts d. This explains why we adopt the oracle in the algorithm.

We have the following two corollaries based on Theorem 5.1, where the notations follow [72, 74].

Corollary 5.1. *For the undirected graph setting, if $\alpha(G_t) \leq \alpha_t$ for $t = 1, \ldots, T$, then setting $\eta = \sqrt{\frac{\log L}{2L\sigma^2 T/\lambda_{min} + d\sum_{t=1}^{T}\alpha_t}}$ gives*

$$\mathcal{R}_T = \mathcal{O}\left(\sqrt{\left(2L\sigma^2 T/\lambda_{min} + d\sum_{t=1}^{T}\alpha_t\right)\log L}\right).$$

Corollary 5.2. *For the directed graph setting, if $\alpha(G_t) \leq \alpha_t$ for $t = 1, \ldots, T$, and supposing that T is large enough so that $\log(1/\gamma) \geq 1$, then setting $\eta = (2L\sigma^2 T/\lambda_{min} + 4d\sum_{t=1}^{T} \alpha_t)^{-\frac{1}{2}}$ gives:*

$$\mathcal{R}_T = \mathcal{O}\left(\sqrt{2L\sigma^2 T/\lambda_{min} + 4d\sum_{t=1}^{T} \alpha_t \log(LdT)}\right).$$

5.2.2 Discussion

Corollaries 5.1 and 5.2 reveal that by properly choosing the learning rate η and the uniform exploration rate γ, the regret of EXP3-LGC-U can be upper bounded by $\mathcal{O}(\sqrt{(L + \alpha(G)d)T \log L})$ in the undirected graph setting, and $\mathcal{O}(\sqrt{(L + \alpha(G)d)T} \log(LdT))$ in the directed graph setting. Compared with state-of-the-art algorithms for adversarial linear contextual bandits, EXP3-LGC-U has tighter regret upper bounds in the extreme case when the feedback graph G_t is a fixed edgeless graph ($\alpha(G) = L$), as [81] shows $\mathcal{O}(5T^{2/3}(Ld \log L)^{1/3})$ for RobustLinEXP3 and $\mathcal{O}(4\sqrt{T} + \sqrt{dLT \log L}(3+ \sqrt{\log T}))$ for RealLinEXP3. It is easily verified that the dependencies on d and T in the regrets of EXP3-LGC-U match with the best existing algorithm RealLinEXP3. Furthermore, the dependence on L of EXP3-LGC-U is matching with the lower bound $\Omega(\sqrt{\alpha(G)T})$ for graphical bandits [71], which improves over that of RealLinEXP3 in general cases. Moreover, our result is also better than algorithms designed for adversarial contextual bandits with arbitrary class of policies [219, 220, 221], which are not capable of guaranteeing an $\mathcal{O}(\sqrt{T})$ regret.

In addition, [81] is different from ours in the following respects: i) loss vector estimator construction, and ii) proof techniques. First, the estimator in [81] is only unbiased w.r.t. $\mathbb{E}_t[\cdot]$ rather than $\mathbb{E}_t[\cdot | X_t]$. Second, their proof is conducted on an auxiliary online learning problem for a fixed context X_0 with L actions (See [81, Lemmas 3 and 4] for details).

5.3 The EXP3-LGC-IX algorithm

In this section, we present another efficient algorithm, EXP3-LGC-IX, for a special class of problems when the support of $\theta_{i,t}$ and X_t is non-negative, and elements of X_t are independent. The motivation for such a setting still comes from the viral marketing problem. Suppose the agent has a questionnaire (context) of some product, which contains true/false questions that are positively weighted. In this case, the answers of users (loss vectors) will be vectors that contain only 0/1 entries. Under the linear payoff assumption, the loss is non-negative. `EXP3-LGC-IX`, which is the abbreviation for "**EXP3** for **L**inear **G**raphical **C**ontextual bandits with **I**mplicit e**X**ploration", has the same regret upper bound for both directed and undirected graph settings, as shown in Section 5.3.1. The proofs for the Claims, Theorems, and Corollaries in this section are deferred to Section 5.5.

Algorithm 5.2: `EXP3-LGC-IX`

```
Require: Learning rate $\eta_t > 0$, implicit exploration rate $\beta_t \in (0, 1)$, covariance
    $\Sigma$, and action set $V$.
for t = 1, ..., T do
    Feedback graph $G_t$ and loss vectors $\{\theta_{i,t}\}_{i \in V}$ are generated but not
    disclosed
    Observe $X_t \sim \mathcal{D}$, and play action $I_t$ drawn according to distribution
    $\pi_t^a(X_t) := (\pi_t^a(1|X_t), \dots, \pi_t^a(L|X_t))$ with
        $\pi_t^a(i|X_t) = \frac{w_t(X_t, i)}{\sum_{j \in V} w_t(X_t, j)},$    (5.7)
    where $w_t(X_t, i) = \frac{1}{L} \exp\left(-\eta_t \sum_{s=1}^{t-1} \left\langle X_t, \hat{\theta}_{i,s} \right\rangle\right)$
    Observe pairs $(i, \ell_t(X_t, i))$ for all $i \in S_{I_t,t}$, disclose feedback graph $G_t$
    Extra observation oracle: observe $\tilde{X}_t \sim \mathcal{D}$ and pairs $(i, \tilde{\ell}_t(\tilde{X}_t, i))$ for all
    $i \in S_{I_t,t}$
    For each $i \in V$, estimate the loss vector $\theta_{i,t}$ as
        $\hat{\theta}_{i,t} = \frac{\mathbb{I}\{i \in S_{I_t,t}\}}{q_t(i|X_t) + \beta_t} \Sigma^{-1} \tilde{X}_t \tilde{\ell}_t(\tilde{X}_t, i),$    (5.8)
    where $q_t(i|X_t) = \pi_t^a(i|X_t) + \sum_{j:j \xrightarrow{t} i} \pi_t^a(j|X_t)$
end for
```

Algorithm 5.2 shows the detailed steps of `EXP3-LGC-IX`, which follows the method of classical `EXP3` and is similar to `EXP3-LGC-U`. The main differences between `EXP3-LGC-IX` and `EXP3-LGC-U` are as follows. First, no explicit uniform exploration mixes with the probability distribution of drawing action (see (5.7)). In this case, for `EXP3-LGC-U` without uniform exploration, only a worse regret upper bound that contains $mas(G)$ rather than $\alpha(G)$ can be proven in the directed graph setting, where $mas(G)$ is the average *maximum acyclic subgraphs number* and $mas(G) \geq \alpha(G)$. This result could be obtained by simply removing the uniform exploration part in the proof of `EXP3-LGC-U` and substituting Lemma 5.3 with [74, Lemma 10]. Second, biased loss vector estimator is adopted (see (5.8)). Similar to `EXP3-IX`, this biased estimator ensures that the loss estimator satisfies the following claim which turns out to be essential for our analysis.

Claim 5.3. *The estimator $\hat{\theta}_{i,t}$ of the loss vector $\theta_{i,t}$ for each $i \in V$ and t satisfies*

$$\begin{aligned}&\mathbb{E}_t\left[\sum_{i\in V}\pi_t^a(i|X_t)\left\langle X_t,\hat{\theta}_{i,t}\right\rangle\middle| X_t\right]\\&=\sum_{i\in V}\pi_t^a(i|X_t)\left\langle X_t,\theta_{i,t}\right\rangle-\beta_t\sum_{i\in V}\frac{\pi_t^a(i|X_t)}{q_t(i|X_t)+\beta_t}\left\langle X_t,\theta_{i,t}\right\rangle.\end{aligned}\tag{5.9}$$

Remark 5.6. Claim 5.3 indicates the loss estimators in `EXP3-LGC-IX` are optimistic. The bias incurred by `EXP3-LGC-IX` can be directly controlled by the implicit exploration rate β_t. This kind of implicit exploration actually has similar effect in controlling the variance of the loss estimators as explicit exploration (e.g., uniform exploration), though the approach is different. Notice that Claim 5.3 does not hold if there is no extra observation oracle (see the proof in Section 5.5.1 for details), which further demonstrates the necessity of the oracle.

5.3.1 Regret analysis for `EXP3-LGC-IX`

The upper bound on the regret of `EXP3-LGC-IX` follows Theorem 5.2, where the proof of Theorem 5.2 is deferred to Section 5.5.2. Notice that a similar higher-order term as that in Remark 5.5 appear in the proof of Theorem 5.2, if the extra observation oracle is not adopted.

Theorem 5.2. *Setting $\beta_t = \sqrt{\log L/(L + \sum_{s=1}^{t-1} Q_s)}$ and*

$\eta_t = \sqrt{\log L/(dL + d\sum_{s=1}^{t-1} Q_s)}$, *the expected regret of* EXP3-LGC-IX *satisfies:*

$$\mathcal{R}_T \leq 2(1+\sqrt{d})\mathbb{E}\left[\sqrt{\left(L + \sum_{t=1}^{T} Q_t\right)\log L}\right], \tag{5.10}$$

for both directed and undirected graph settings, where $Q_t = 2\alpha(G_t)\log\left(1 + \frac{\lceil L^2/\beta_t \rceil + L}{\alpha(G_t)}\right) + 2$.

Based on Theorem 5.2, we have the following corollary.

Corollary 5.3. *Suppose* $\alpha(G_t) \leq \alpha_t$ *for* $t = 1, \ldots T$, *the regret of* EXP3-LGC-IX *satisfies*

$$\mathcal{R}_T = \mathcal{O}\left(\sqrt{\sum_{t=1}^{T} \alpha_t d \log L \log{(LT)}}\right),$$

for both directed and undirected graph settings.

Corollary 5.3 reveals that by adopting the learning rate η_t and the implicit exploration rate β_t adaptively, the regret of EXP3-LGC-IX can be upper bounded by $\mathcal{O}(\sqrt{\alpha(G)dT\log L\log(LT)})$ for both directed and undirected graph settings. This result indicates that EXP3-LGC-IX captures the benefits of both contexts and side observations, as discussed in Section 5.2.2. The EXP3-LGC-IX algorithm cannot handle negative losses due to the following two reasons. First, if the losses are negative, Claim 5.3 does not hold. Second, although we can flip the sign of β_t according to the sign of the loss vector to guarantee the optimism of the loss estimator, the graph-theoretic result (e.g., [82, Lemma 2]) cannot be applied as β_t is required to be positive.

5.4 Proofs in Section 5.2

5.4.1 Proof of Claim 5.1

Proof. By plugging (5.4) into $\mathbb{E}_t\left[\hat{\theta}_{i,t}\middle| X_t\right]$, we have that

$$\begin{aligned}\mathbb{E}_t\left[\hat{\theta}_{i,t}\middle| X_t\right] &= \mathbb{E}_t\left[\frac{\mathbb{I}\{i\in S_{I_t}\}}{q_t(i|X_t)}\Sigma^{-1}\tilde{X}_t\tilde{\ell}_t(\tilde{X}_t,i)\middle| X_t\right]\\ &= \mathbb{E}_t\left[\frac{\mathbb{I}\{i\in S_{I_t}\}}{q_t(i|X_t)}\Sigma^{-1}\tilde{X}_t\tilde{X}_t^\top\theta_{i,t}\middle| X_t\right]\\ &\overset{\text{(a)}}{=} \mathbb{E}_t\left[\frac{\mathbb{I}\{i\in S_{I_t}\}}{q_t(i|X_t)}\middle| X_t\right]\Sigma^{-1}\mathbb{E}\left[\tilde{X}_t\tilde{X}_t^\top\right]\theta_{i,t},\end{aligned}$$

where step (a) uses the fact that $\mathbb{E}_t\left[\theta_{i,t}\middle| X_t\right] = \mathbb{E}_t\left[\theta_{i,t}\right] = \theta_{i,t}$, and $\tilde{X}_t$ is independent of $\mathcal{F}_{t-1}$, X_t, and $\theta_{i,t}$. Notice the following facts,

$$\mathbb{E}_t\left[\frac{\mathbb{I}\{i\in S_{I_t}\}}{q_t(i|X_t)}\middle| X_t\right] = \sum_{j:i\in S_{j,t}}\frac{\pi_t^a(j|X_t)}{q_t(i|X_t)} = 1, \text{ and } \mathbb{E}\left[\tilde{X}_t\tilde{X}_t^\top\right] = \Sigma.$$

We conclude that $\mathbb{E}_t\left[\hat{\theta}_{i,t}\middle| X_t\right] = \theta_{i,t}$. □

5.4.2 Proof of Claim 5.2

Proof.

$$\begin{aligned}&\mathbb{E}\left[\sum_{i\in V}\left(\pi_t^a(i|X_t)-\pi_T(i|X_t)\right)\left\langle X_t,\hat{\theta}_{i,t}\right\rangle\right]\\ &\overset{\text{(a)}}{=} \mathbb{E}\left[\mathbb{E}_t\left[\sum_{i\in V}\left(\pi_t^a(i|X_t)-\pi_T(i|X_t)\right)\left\langle X_t,\hat{\theta}_{i,t}\right\rangle\middle| X_t\right]\right]\\ &= \mathbb{E}\left[\sum_{i\in V}\left(\pi_t^a(i|X_t)-\pi_T(i|X_t)\right)\left\langle X_t,\mathbb{E}_t\left[\hat{\theta}_{i,t}\middle| X_t\right]\right\rangle\right]\\ &\overset{\text{(b)}}{=} \mathbb{E}\left[\sum_{i\in V}\left(\pi_t^a(i|X_t)-\pi_T(i|X_t)\right)\left\langle X_t,\theta_{i,t}\right\rangle\right],\end{aligned}$$

where step (a) uses the law of total expectation and step (b) uses Claim 5.1.

□

5.4.3 Proof of Lemma 5.1

Proof. As mentioned before, Claim 5.2 enables us to adopt the techniques similar to the ones used to originally analyze `EXP3` in [217]. We introduce $W_t(x) = \sum_{i\in V} w_t(x,i)$ for convenience, where $w_t(x,i)$ is defined in (5.2). With the assumption that $|\eta\langle X_t, \hat{\theta}_{i,t}\rangle| \leq 1$, the following result holds for each $t = 1, \dots, T$,

$$\begin{aligned}
&\log \frac{W_{t+1}(X_t)}{W_t(X_t)} \\
&= \log\left(\sum_{i\in V} \frac{w_{t+1}(X_t,i)}{W_t(X_t)}\right) \\
&= \log\left(\sum_{i\in V} \frac{w_t(X_t,i)}{W_t(X_t)} \cdot e^{-\eta\left\langle X_t, \hat{\theta}_{i,t}\right\rangle}\right) \\
&\overset{\text{(a)}}{=} \log\left(\sum_{i\in V} \frac{\pi_t^a(i|X_t) - \gamma/L}{1-\gamma} \cdot e^{-\eta\left\langle X_t, \hat{\theta}_{i,t,}\right\rangle}\right) \\
&\overset{\text{(b)}}{\leq} \log\left(\sum_{i\in V} \frac{\pi_t^a(i|X_t) - \gamma/L}{1-\gamma} \left(1 - \eta\left\langle X_t, \hat{\theta}_{i,t}\right\rangle + \eta^2 \left\langle X_t, \hat{\theta}_{i,t}\right\rangle^2\right)\right) \\
&= \log\left(1 + \sum_{i\in V} \frac{\pi_t^a(i|X_t) - \gamma/L}{1-\gamma} \left(-\eta\left\langle X_t, \hat{\theta}_{i,t}\right\rangle + \eta^2 \left\langle X_t, \hat{\theta}_{i,t}\right\rangle^2\right)\right) \\
&\overset{\text{(c)}}{\leq} \sum_{i\in V} \frac{\pi_t^a(i|X_t)}{1-\gamma} \left(-\eta\left\langle X_t, \hat{\theta}_{i,t}\right\rangle + \eta^2 \left\langle X_t, \hat{\theta}_{i,t}\right\rangle^2\right) \\
&\qquad + \frac{\eta\gamma}{L(1-\gamma)} \sum_{i\in V} \left\langle X_t, \hat{\theta}_{i,t}\right\rangle,
\end{aligned} \tag{5.11}$$

where equality (a) uses the definition of $\pi_t^a(i|X_t)$ in (5.3), in step (a) the inequality $e^{-z} \leq 1 - z + z^2$ that holds for $z \geq -1$ is used, and in step (b) the inequality $\log(1+z) \leq z$ that holds for $z > -1$ is used.

The key to this proof is in the following. By drawing X from the distribution $\mathcal{D}$ that is independent of the entire interaction history $\mathcal{F}_T$, and substituting X_t with X, we have that

$$\mathbb{E}\left[\log \frac{W_{t+1}(X_t)}{W_t(X_t)}\right] = \mathbb{E}\left[\log \frac{W_{t+1}(X)}{W_t(X)}\right].$$

This is because X_t and X are i.i.d., and for each term $\log(W_{t+1}(X_t)/W_t(X_t))$, only X_t is substituted with X while $X_1, \dots, X_{t-1}$ remain unchanged. Repeatedly, we apply this step to $\mathbb{E}\left[\log \frac{W_{t+1}(X_t)}{W_t(X_t)}\right]$ for each t, which leads to the following lower bound,

$$
\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \log \frac{W_{t+1}(X_t)}{W_t(X_t)}\right] &= \mathbb{E}\left[\sum_{t=1}^{T} \log \frac{W_{t+1}(X)}{W_t(X)}\right] \\
&= \mathbb{E}\left[\log \frac{W_{T+1}(X)}{W_1(X)}\right] \\
&\overset{\text{(a)}}{\geq} \mathbb{E}\left[\log \frac{w_{T+1}(X, \pi_T(X))}{W_1(X)}\right] \\
&\overset{\text{(b)}}{=} \mathbb{E}\left[-\eta \sum_{t=1}^{T} \left\langle X, \hat{\theta}_{\pi_T(X),t} \right\rangle - \log L\right] \\
&\overset{\text{(c)}}{=} \mathbb{E}\left[-\eta \sum_{t=1}^{T} \left\langle X_t, \hat{\theta}_{\pi_T(X_t),t} \right\rangle - \log L\right] \\
&= \mathbb{E}\left[-\eta \sum_{t=1}^{T} \sum_{i \in V} \pi_T(i|X_t) \left\langle X_t, \hat{\theta}_{i,t} \right\rangle - \log L\right], \quad (5.12)
\end{aligned}
$$

where inequality (a) is due to the fact that $W_{T+1}(X) \geq w_{T+1}(X, \pi_T(X))$, step (b) is derived from the definition of $w_{T+1}(X, \pi_T(X))$ and the fact that $\log(W_1(X)) = L$, and step (c) is realized by substituting X with X_t in each of $\left\langle X, \hat{\theta}_{\pi_T(X),t} \right\rangle$ as X_t and X are i.i.d. Combining the upper bound in (5.11) and the lower bound in (5.12) gives

$$
\begin{aligned}
&\mathbb{E}\left[-\eta \sum_{t=1}^{T} \sum_{i \in V} \pi_T(i|X_t) \left\langle X_t, \hat{\theta}_{i,t} \right\rangle - \log L\right] \\
&\quad \leq \mathbb{E}\left[\sum_{t=1}^{T} \sum_{i \in V} \frac{\pi_t^a(i|X_t)}{1-\gamma} \left(-\eta \left\langle X_t, \hat{\theta}_{i,t} \right\rangle + \eta^2 \left\langle X_t, \hat{\theta}_{i,t} \right\rangle^2\right)\right. \\
&\quad \left. + \frac{\eta\gamma}{L(1-\gamma)} \sum_{i \in V} \left\langle X_t, \hat{\theta}_{i,t} \right\rangle\right].
\end{aligned}
$$

Reordering and multiplying both sides by $\frac{1-\gamma}{\eta}$ gives

$$
\mathbb{E}\left[\sum_{t=1}^{T} \sum_{i \in V} (\pi_t^a(i|X_t) - \pi_T(i|X_t)) \left\langle X_t, \hat{\theta}_{i,t} \right\rangle\right]
$$

$$\leq \frac{(1-\gamma)\log L}{\eta} + \eta \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i\in V} \pi_t^a(i|X_t)\left\langle X_t, \hat{\theta}_{i,t}\right\rangle^2\right]$$
$$+ \gamma \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i\in V}\left(\frac{1}{L} - \pi_T(i|X_t)\right)\left\langle X_t, \hat{\theta}_{i,t}\right\rangle\right].$$

Furthermore, combining Claim 5.2 with the fact that

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i\in V}\left(\frac{1}{L} - \pi_T(i|X_t)\right)\left\langle X_t, \hat{\theta}_{i,t}\right\rangle\right]$$
$$= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i\in V}\left(\frac{1}{L} - \pi_T(i|X_t)\right)\left\langle X_t, \theta_{i,t}\right\rangle\right] \leq 2T,$$

and $(1-\gamma) \leq 1$, we conclude with

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i\in V}(\pi_t^a(i|X_t) - \pi_T(i|X_t))\left\langle X_t, \hat{\theta}_{i,t}\right\rangle\right]$$
$$\leq \frac{\log L}{\eta} + 2\gamma T + \eta \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i\in V}\pi_t^a(i|X_t)\left\langle X_t, \hat{\theta}_{i,t}\right\rangle^2\right].$$

Since the above steps hold for any $\pi_T \in \Pi$, we have that

$$\mathcal{R}_T \leq \frac{\log L}{\eta} + 2\gamma T + \eta \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i\in V}\pi_t^a(i|X_t)\left\langle X_t, \hat{\theta}_{i,t}\right\rangle^2\right].$$

□

5.4.4 Proof of Theorem 5.1

Before presenting the proof of Theorem 5.1, we restate the following two graph-theoretic results from [71, 74] for convenience.

Lemma 5.2 (Lemma 10 in [74]). *Let G_t be an undirected graph. For any distribution π over V,*

$$\sum_{i\in V}\frac{\pi(i)}{\pi(i) + \sum_{j:j\xrightarrow{t}i}\pi(j)} \leq \alpha(G_t).$$

Lemma 5.3 (Lemma 5 in [73]). *Let G_t be a directed graph and π be any probability distribution over V. Assume that $\pi(i) \geq \epsilon$ for all $i \in V$ for some constant $0 < \epsilon < \frac{1}{2}$. Then,*

$$\sum_{i \in V} \frac{\pi(i)}{\pi(i) + \sum_{j:j \xrightarrow{t} i} \pi(j)} \leq 4\alpha(G_t) \log \frac{4L}{\alpha(G_t)\epsilon}.$$

Proof of Theorem 5.1. Using Lemma 5.1, we are left to upper bound the term $\mathbb{E}\left[\sum_{t=1}^{T} \sum_{i \in V} \pi_t^a(i|X_t) \left\langle X_t, \hat{\theta}_{i,t} \right\rangle^2\right]$. Substituting (5.4) into this term yields,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i \in V} \pi_t^a(i|X_t) \left\langle X_t, \hat{\theta}_{i,t} \right\rangle^2\right] \\
&= \mathbb{E}\left[\sum_{i \in V} \pi_t^a(i|X_t) \frac{\mathbb{I}\{i \in S_{I_t,t}\} \tilde{l}_t^2(\tilde{X}_t, i)}{q_t^2(i|X_t)} X_t^\top \Sigma^{-1} \tilde{X}_t \tilde{X}_t^\top \Sigma^{-1} X_t\right] \\
&\overset{\text{(a)}}{\leq} \mathbb{E}\left[\sum_{i \in V} \pi_t^a(i|X_t) \frac{\mathbb{I}\{i \in S_{I_t,t}\}}{q_t^2(i|X_t)} X_t^\top \Sigma^{-1} \tilde{X}_t \tilde{X}_t^\top \Sigma^{-1} X_t\right] \qquad (5.13) \\
&\overset{\text{(b)}}{=} \mathbb{E}\left[\sum_{i \in V} \underbrace{\mathbb{E}_t\left[\frac{\mathbb{I}\{i \in S_{I_t,t}\}}{q_t^2(i|X_t)} \middle| X_t\right]}_{A} \pi_t^a(i|X_t) X_t^\top \Sigma^{-1} \tilde{X}_t \tilde{X}_t^\top \Sigma^{-1} X_t\right],
\end{aligned}$$

where the step (a) is due to the fact that $\tilde{l}_t^2(\tilde{X}_t, i) \leq 1$, and step (b) uses the law of total expectation. We have the following result for term A:

$$A = \sum_{j: i \in S_{j,t}} \frac{\pi(j|X_t)}{q_t^2(i|X_t)} = \frac{q_t(i|X_t)}{q_t^2(i|X_t)} = \frac{1}{q_t(i|X_t)}.$$

According to Lemmas 5.2 and 5.3, we know that

$$\sum_{i \in V} \frac{\pi_t^a(i|X_t)}{q_t(i|X_t)} \leq Q_t,$$

where Q_t is $\alpha(G_t)$ for undirected graph setting and $4\alpha(G_t) \log(4L^2/(\alpha(G_t)\gamma))$

for directed graph setting. Also, Q_t is independent of X_t and $\tilde{X}_t$. Thus,

$$\begin{aligned}\mathbb{E}\left[\sum_{i\in V}\pi_t^a(i|X_t)\left\langle X_t,\hat{\theta}_{i,t}\right\rangle^2\right] &\le \mathbb{E}\left[Q_t\right]\mathbb{E}\left[X_t^\top\Sigma^{-1}\tilde{X}_t\tilde{X}_t^\top\Sigma^{-1}X_t\right]\\ &= \mathbb{E}\left[Q_t\right]\mathbb{E}\left[\mathrm{tr}(\Sigma^{-1}\tilde{X}_t\tilde{X}_t^\top\Sigma^{-1}X_tX_t^\top)\right]\\ &= d\mathbb{E}\left[Q_t\right].\end{aligned}$$

In addition, we must ensure that $\eta\left|\left\langle X_t,\hat{\theta}_{i,t}\right\rangle\right|\le 1$ for all $t=1,\dots,T$:

$$\left|\left\langle X_t,\hat{\theta}_{i,t}\right\rangle\right| = \frac{\mathbb{I}\{i\in S_{i,t}\}}{q_t(i|X_t)}\left|X_t^\top\Sigma^{-1}\tilde{X}_t\tilde{l}_t(\tilde{X}_t,i)\right| \le \frac{L\sigma^2}{\lambda_{\min}\gamma},$$

where we use the fact that $q_t(i|X_t)\ge\pi_t^a(i|X_t)\ge\frac{\gamma}{L}$, $|\tilde{l}_t(\tilde{X}_t,i)|\le 1$, and $\left|X_t^\top\Sigma^{-1}\tilde{X}_t\right|\le\frac{\sigma^2}{\lambda_{\min}}$. Choosing $\gamma=\frac{\eta L\sigma^2}{\lambda_{\min}}$ guarantees $\eta\left|\left\langle X_t,\hat{\theta}_{i,t}\right\rangle\right|\le 1$, which concludes the proof. □

5.4.5 Proofs of Corollaries 5.1 and 5.2

Proof of Corollary 5.1. Given the fact $Q_t=\alpha(G_t)$ in Theorem 5.1, and assuming $\alpha(G_t)\le\alpha_t$ for $t=1,\dots,T$, we conclude that

$$\mathcal{R}_T=\mathcal{O}\left(\sqrt{\left(2L\sigma^2T/\lambda_{\min}+d\sum_{t=1}^T\alpha_t\right)\log L}\right),$$

by setting $\eta=\sqrt{\frac{\log L}{2L\sigma^2T/\lambda_{\min}+d\sum_{t=1}^T\alpha_t}}$. □

Proof of Corollary 5.2. Define $f(z)=4z\log(4L^2/(z\gamma))$ for $z\le L$, and we have that

$$f'(z)=4\log\frac{4L^2}{\gamma}-4\log z-4.$$

Notice that $4\log(4L^2)>4\log z$, and so $f(z)$ is an increasing function as long as $\log(1/\gamma)\ge 1$. If $\alpha(G_t)\le\alpha_t$ for $t=1,\dots,T$, the following result holds if $\log(1/\gamma)\ge 1$,

$$\mathbb{E}\left[4\alpha(G_t)\log\frac{4L^2}{\alpha(G_t)\gamma}\right]\le 4\alpha_t\log\frac{4L^2}{\alpha_t\gamma}.$$

By choosing $\eta=\sqrt{1/(L\sigma^2T/\lambda_{\min}+4d\sum_{t=1}^T\alpha_t)}$ and $\gamma=\frac{\eta L\sigma^2}{\lambda_{\min}}$, we conclude that

$$\mathcal{R}_T = \mathcal{O}\left(\sqrt{\left(\frac{L\sigma^2}{\lambda_{\min}}T + 4d\sum_{t=1}^{T}\alpha_t\right)\log(LdT)}\right).$$

□

5.5 Proofs for Section 5.3

5.5.1 Proof of Claim 5.3

Proof.

$$\begin{aligned}
&\mathbb{E}_t\left[\sum_{i\in V}\pi_t^a(i|X_t)\left\langle X_t,\hat{\theta}_{i,t}\right\rangle\middle| X_t\right]\\
&=\sum_{i\in V}\pi_t^a(i|X_t)\frac{1}{q_t(i|X_t)+\beta_t}X_t^T\Sigma^{-1}\mathbb{E}_t\left[\mathbb{I}\{i\in S_{I_t,t}\}\tilde{X}_t\tilde{X}_t^\top\middle| X_t\right]\theta_{i,t}\\
&\overset{\text{(a)}}{=}\sum_{i\in V}\pi_t^a(i|X_t)\frac{q_t(i|X_t)}{q_t(i|X_t)+\beta_t}\langle X_t,\theta_{i,t}\rangle\\
&=\sum_{i\in V}\pi_t^a(i|X_t)\langle X_t,\theta_{i,t}\rangle-\beta_t\sum_{i\in V}\frac{\pi_t^a(i|X_t)}{q_t(i|X_t)+\beta_t}\langle X_t,\theta_{i,t}\rangle,
\end{aligned}$$

where the equality (a) holds because $\mathbb{I}\{i\in S_{I_t,t}\}$ and $\tilde{X}_t$ are independent. □

5.5.2 Proof of Theorem 5.2

To prove Theorem 5.2, we need the graph-theoretic result from [82, Kocak et al., 2014] which is restated here.

Lemma 5.4 (Lemma 2 in [82]). *Let G_t be a directed or undirected graph with vertex set $V := \{1,\ldots,L\}$. Let $\alpha(G_t)$ be the independence number of G_t and π be a distribution over V. Then,*

$$\sum_{i\in V}\frac{\pi(i)}{c+\pi(i)+\sum_{j:j\xrightarrow{t}i}\pi(j)}\leq 2\alpha(G_t)\log\left(1+\frac{\lceil L^2/c\rceil+L}{\alpha(G_t)}\right)+2,$$

where c is a positive constant.

Proof of Theorem 5.2. We start by recalling the notation $w_t(x,i) = \exp(-\eta_t \sum_{s=1}^{t-1} \left\langle x, \hat{\theta}_{i,s} \right\rangle)/L$ in (5.8), and introducing $W_t(x) = \sum_{i \in V} w_t(x,i)$ and $W'_t(x) = \sum_{i \in V} \exp\left(-\eta_{t-1} \sum_{s=1}^{t-1} \left\langle x, \hat{\theta}_{i,s} \right\rangle\right)/L$. The proof follows [82] with some additional techniques.

$$
\begin{aligned}
&\frac{1}{\eta_t} \log \frac{W'_{t+1}(X_t)}{W_t(X_t)} \\
&= \frac{1}{\eta_t} \log \left(\sum_{i \in V} \frac{w_t(X_t, i)}{W_t(X_t)} e^{-\eta_t \left\langle X_t, \hat{\theta}_{i,t} \right\rangle} \right) \\
&= \frac{1}{\eta_t} \log \left(\sum_{i \in V} \pi_t^a(i|X_t) e^{-\eta_t \left\langle X_t, \hat{\theta}_{i,t} \right\rangle} \right) \\
&\overset{\text{(a)}}{\leq} \frac{1}{\eta_t} \log \left(\sum_{i \in V} \pi_t^a(i|X_t) \left(1 - \eta_t \left\langle X_t, \hat{\theta}_{i,t} \right\rangle + \frac{1}{2}\eta_t^2 \left\langle X_t, \hat{\theta}_{i,t} \right\rangle^2 \right) \right) \\
&= \frac{1}{\eta_t} \log \left(1 + \sum_{i \in V} \pi_t^a(i|X_t) \left(-\eta_t \left\langle X_t, \hat{\theta}_{i,t} \right\rangle + \frac{1}{2}\eta_t^2 \left\langle X_t, \hat{\theta}_{i,t} \right\rangle^2 \right) \right) \\
&\overset{\text{(b)}}{\leq} \frac{1}{\eta_t} \sum_{i \in V} \pi_t^a(i|X_t) \left(-\eta_t \left\langle X_t, \hat{\theta}_{i,t} \right\rangle + \frac{1}{2}\eta_t^2 \left\langle X_t, \hat{\theta}_{i,t} \right\rangle^2 \right), \qquad (5.14)
\end{aligned}
$$

where step (a) uses the inequality $\exp(-z) \leq 1 - z + z^2/2$ that holds for $z \geq 0$ and step (b) uses the inequality $\log(1+z) \leq z$ that holds for all $z > -1$. Notice that

$$
\begin{aligned}
W_{t+1}(X_t) &= \sum_{i \in V} \frac{1}{L} e^{-\eta_{t+1} \sum_{s=1}^{t} \left\langle X_t, \hat{\theta}_{i,t} \right\rangle} \\
&= \sum_{i \in V} \frac{1}{L} \left(e^{-\eta_t \sum_{s=1}^{t} \left\langle X_t, \hat{\theta}_{i,t} \right\rangle} \right)^{\frac{\eta_{t+1}}{\eta_t}} \\
&\overset{\text{(a)}}{\leq} \left(\frac{1}{L} \sum_{i \in V} e^{-\eta_t \sum_{s=1}^{t} \left\langle X_t, \hat{\theta}_{i,t} \right\rangle} \right)^{\frac{\eta_{t+1}}{\eta_t}} \\
&= \left(W'_{t+1}(X_t) \right)^{\frac{\eta_{t+1}}{\eta_t}}, \qquad (5.15)
\end{aligned}
$$

where step (a) uses Jensen's inequality for the concave function $z^{\frac{\eta_{t+1}}{\eta_t}}$ for all $z \in \mathbb{R}$ as η_t is a decreasing sequence. Taking the $\log(\cdot)$ on both side of (5.15), we have that

$$
\frac{1}{\eta_t} \log \frac{W'_{t+1}(X_t)}{W_t(X_t)} \geq \frac{\log W_{t+1}(X_t)}{\eta_{t+1}} - \frac{\log W_t(X_t)}{\eta_t}.
$$

The following fact that can be easily interpreted using the same techniques as Lemma 5.1:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\left(\frac{\log W_{t+1}(X_t)}{\eta_{t+1}}-\frac{\log W_t(X_t)}{\eta_t}\right)\right]\\
&=\mathbb{E}\left[\frac{\log W_{T+1}(X)}{\eta_{T+1}}-\frac{\log W_1(X)}{\eta_1}\right]\\
&\geq\mathbb{E}\left[\frac{\log w_{T+1}(X,\pi_T(X))}{\eta_{T+1}}-\frac{\log W_1(X)}{\eta_1}\right]\\
&=-\mathbb{E}\left[\frac{\log L}{\eta_{T+1}}\right]-\mathbb{E}\left[\sum_{t=1}^{T}\left\langle X,\hat{\theta}_{\pi_T(X),t}\right\rangle\right]\\
&=-\mathbb{E}\left[\frac{\log L}{\eta_{T+1}}\right]-\mathbb{E}\left[\sum_{t=1}^{T}\left\langle X_t,\hat{\theta}_{\pi_T(X_t),t}\right\rangle\right], \qquad (5.16)
\end{aligned}$$

where $X \sim D$ is independent from the whole interaction history $\mathcal{F}_t$. Wrapping up above steps in (5.14) and (5.16) and applying the Claim 5.3, the $\theta_{i,t}$ is an optimistic estimator that $\mathbb{E}\left[\sum_{t=1}^{T}\left\langle X_t,\hat{\theta}_{\pi_T(X_t),t}\right\rangle\right] \leq$ $\mathbb{E}\left[\sum_{t=1}^{T}\left\langle X_t,\theta_{\pi_T(X_t),t}\right\rangle\right]$, we have that

$$\begin{aligned}
&-\mathbb{E}\left[\frac{\log L}{\eta_{T+1}}\right]-\mathbb{E}\left[\sum_{t=1}^{T}\left\langle X_t,\theta_{\pi_T(X_t),t}\right\rangle\right]\\
&\leq\mathbb{E}\left[\sum_{i=1}^{T}\sum_{i\in V}\pi_t^a(i|X_t)\left(-\left\langle X_t,\hat{\theta}_{i,t}\right\rangle+\frac{1}{2}\eta_t\left\langle X_t,\hat{\theta}_{i,t}\right\rangle^2\right)\right]. \qquad (5.17)
\end{aligned}$$

Notice that

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i\in V}\pi_t^a(i|X_t)\left\langle X_t,\hat{\theta}_{i,t}\right\rangle\right]\\
&=\mathbb{E}\left[\mathbb{E}_t\left[\sum_{i\in V}\pi_t^a(i|X_t)\left\langle X_t,\hat{\theta}_{i,t}\right\rangle\middle| X_t\right]\right]\\
&\overset{(a)}{=}\mathbb{E}\left[\sum_{i\in V}\pi_t^a(i|X_t)\left\langle X_t,\theta_{i,t}\right\rangle-\beta_t\sum_{i\in V}\frac{\pi_t^a(i|X_t)}{q_t(i|X_t)+\beta_t}\left\langle X_t,\theta_{i,t}\right\rangle\right]\\
&\overset{(b)}{\geq}\mathbb{E}\left[\sum_{i\in V}\pi_t^a(i|X_t)\left\langle X_t,\theta_{i,t}\right\rangle-\beta_t Q_t\right], \qquad (5.18)
\end{aligned}$$

where step (a) is due to Claim 5.3, and step (b) uses Lemma 5.4 and

$\langle X_t, \theta_{i,t}\rangle \in [0,1]$. Also,

$$\begin{aligned}
&\mathbb{E}\left[\eta_t \sum_{i\in V} \pi_t^a(i|X_t) \left\langle X_t, \hat{\theta}_{i,t}\right\rangle^2\right] = \mathbb{E}\left[\mathbb{E}_t\left[\eta_t \sum_{i\in V} \pi_t^a(i|X_t) \left\langle X_t, \hat{\theta}_{i,t}\right\rangle^2 \middle| X_t\right]\right]\\
&\le \mathbb{E}\left[\mathbb{E}\left[\eta_t\right] \sum_{i\in V} \frac{\pi_t^a(i|X_t)}{(q_t(i|X_t)+\beta_t)^2} X_t^\top \Sigma^{-1} \mathbb{E}_t\left[\mathbb{I}\{i\in S_{I_t,t}\}\tilde{X}_t\tilde{X}_t^\top \middle| X_t\right] \Sigma^{-1} X_t\right]\\
&= \mathbb{E}\left[\mathbb{E}\left[\eta_t\right] \sum_{i\in V} \frac{\pi_t^a(i|X_t) q_t(i|X_t)}{(q_t(i|X_t)+\beta_t)^2} X_t^\top \Sigma^{-1} X_t\right]\\
&= \mathbb{E}\left[\eta_t \sum_{i\in V} \frac{\pi_t^a(i|X_t) q_t(i|X_t)}{(q_t(i|X_t)+\beta_t) q_t(i|X_t)} X_t^\top \Sigma^{-1} X_t\right]\\
&\overset{\text{(a)}}{\le} \mathbb{E}\left[\eta_t Q_t \mathrm{tr}(\Sigma^{-1} X_t X_t^\top)\right] \le \mathbb{E}\left[\eta_t Q_t\right] d,
\end{aligned} \tag{5.19}$$

where step (a) uses Lemma 5.4. By reordering the results in Eqs. (5.17), (5.18), and (5.19), we have that

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} (\pi_t^a(i|X_t) - \pi_T(i|X_t)) \langle X_t, \theta_t\rangle\right]\\
&\le \mathbb{E}\left[\frac{\log L}{\eta_{t+1}}\right] + \sum_{t=1}^{T} \mathbb{E}\left[\beta_t Q_t\right] + \frac{d}{2} \sum_{i=1}^{T} \mathbb{E}\left[\eta_t Q_t\right].
\end{aligned} \tag{5.20}$$

Plugging in η_t and β_t and using Lemma 3.5 in [224], the result in (5.20) becomes

$$\mathcal{R}_T \le 2(1+\sqrt{d})\mathbb{E}\left[\sqrt{\left(L + \sum_{t=1}^{T} Q_t\right) \log L}\right],$$

which holds for all $\pi_T \in \Pi$. □

5.5.3 Proof of Corollary 5.3

Proof. Notice that $x \log(1 + a/x)$ is an increasing function of $x \in (0, \infty]$ for any $a > 0$, and thus

$$Q_t \le 2\alpha_t \log\left(1 + \frac{\lceil L^2/\beta_t \rceil + L}{\alpha_t}\right) + 2,$$

if $\alpha(G_t) \leq \alpha_t$ for $t = 1, \dots T$. Using the fact

$$\log\left(1 + \frac{\lceil L^2/\beta_t \rceil + L}{\alpha_t}\right) \leq \log\left(1 + \frac{\lceil L^2\sqrt{tL/\log L} \rceil + L}{\alpha_t}\right) = \mathcal{O}(\log(LT)),$$

we conclude that

$$\mathcal{R}_T = \mathcal{O}\left(\sqrt{\sum_{t=1}^{T} \alpha_t d \log L \log LT}\right),$$

for both directed and undirected graph settings. □

Part II

Joint Community Detection and Phase Synchronization

CHAPTER 6

MULTI-FREQUENCY JOINT COMMUNITY DETECTION AND PHASE SYNCHRONIZATION

6.1 Preliminaries

6.1.1 Notations

Throughout Chapter 6, we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$, and $\mathbb{I}\{\cdot\}$ to denote the indicator function. The uppercase and lowercase letters in boldface are used to represent matrices and vectors, while normal letters are reserved for scalars. $\|\boldsymbol{X}\|_{\mathrm{F}}$ and $\mathrm{Tr}(\boldsymbol{X})$ denote the Frobenius norm and the trace of matrix $\boldsymbol{X}$, and $\|\boldsymbol{v}\|_2$ denotes the ℓ_2 norm of the vector $\boldsymbol{v}$. The transpose and conjugate transpose of a matrix $\boldsymbol{X}$ (resp. a vector $\boldsymbol{x}$) are denoted by $\boldsymbol{X}^\top$ and $\boldsymbol{X}^{\mathsf{H}}$ (resp. $\boldsymbol{x}^\top$ and $\boldsymbol{x}^{\mathsf{H}}$), respectively. An $m \times n$ matrix of all zeros is denoted by $\mathbf{0}_{m\times n}$ (or $\mathbf{0}$, for brevity). An identity matrix of size $n \times n$ is defined as $\boldsymbol{I}_n$. The complex conjugate of x is denoted by $\overline{x}$. The inner product $\langle \cdot, \cdot \rangle$ between two scalars, vectors, and matrices are $\langle x, y \rangle = \overline{x}y$, $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^{\mathsf{H}}\boldsymbol{y}$, and $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = \mathrm{Tr}(\boldsymbol{X}^{\mathsf{H}}\boldsymbol{Y})$, respectively. In terms of indexing, (i, j)th entry of $\boldsymbol{X}$ is denoted by $\boldsymbol{X}_{ij}$, and ith entry of $\boldsymbol{x}$ is denoted by $\boldsymbol{x}_i$. $\boldsymbol{X}_{i,\cdot}$ (resp. $\boldsymbol{X}_{\cdot,j}$) is used to denote ith row (resp. jth column) of $\boldsymbol{X}$. We use $\boldsymbol{X}_{i,j:}$ (resp. $\boldsymbol{X}_{i:,j}$) to denote the segment of the ith row (resp. jth column) from the jth entry (resp. ith entry) to the end, and $\boldsymbol{x}_{i:}$ to denote the segment from ith entry to the end. In addition, the sub-matrix of $\boldsymbol{X}$ from the ith row and jth column to the end is denoted by $\boldsymbol{X}_{i:,j:}$. Lastly, we use $\mathcal{O}$ and Θ to denote the usual Big-O and Big-Theta notations. The notations are summarized in Table 6.1.

Table 6.1: Notation table.

[n]	Set of first n positive integers: $1, \ldots, n$.
$\mathbb{I}\{\cdot\}$	Indicator function.
$\boldsymbol{X}$, $\boldsymbol{x}$, x	Matrix, vector, scalar.
$\boldsymbol{X}^\top$, $\boldsymbol{X}^\mathsf{H}$	Transpose, conjugate transpose.
$\overline{x}$	Complex conjugate.
$\langle \cdot, \cdot \rangle$	Inner product.
$\lVert \cdot \rVert_\mathrm{F}$	Frobenius norm of a matrix.
$\lVert \cdot \rVert_2$	ℓ_2 norm of a vector.
$\mathbf{0}_{m\times n}$ (or $\mathbf{0}$)	All zero matrix of size $m \times n$.
$\boldsymbol{I}_n$	Identity matrix of size $n \times n$.
$\boldsymbol{X}_{ij}$	the (i, j)th entry of $\boldsymbol{X}$.
$\boldsymbol{x}_i$	the ith entry of $\boldsymbol{x}$.
$\boldsymbol{X}_{i,\cdot}$ ($\boldsymbol{X}_{\cdot,j}$)	the ith row (the jth column) of $\boldsymbol{X}$.
$\boldsymbol{X}_{i,j:}$ ($\boldsymbol{X}_{i:,j}$)	Segment of the ith row (the jth column) from the jth entry (the ith entry) to the end of the row (the column).
$\boldsymbol{x}_{i:}$	Segment of the vector $\boldsymbol{x}$ from the ith entry to the end.
$\boldsymbol{X}_{i:,j:}$	Sub-matrix of $\boldsymbol{X}$ from the ith row and the jth column to the end.
$\mathcal{O}$	Big-O notation.
Θ	Big-Theta notation.

6.1.2 Definitions

Definition 6.1 (QR factorization)**.** *Given $\boldsymbol{X} \in \mathbb{C}^{m\times n}$, a QR factorization of $\boldsymbol{X}$ satisfies*

$$\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{R},$$

where $\boldsymbol{Q} \in \mathbb{C}^{m\times m}$ is a unitary matrix, and $\boldsymbol{R} \in \mathbb{C}^{m\times n}$ is an upper triangular matrix.

Such factorization always exists for any $\boldsymbol{X}$. The most common methods for computing the QR factorization are Gram-Schmidt process [225] and Householder transformation [226].

Definition 6.2 (Column-pivoted QR factorization)**.** *Let $\boldsymbol{X} \in \mathbb{C}^{m\times n}$ with $m \leq n$ has rank m. The column-pivoted QR factorization of $\boldsymbol{X}$ is the factorization*

$$\boldsymbol{X}\boldsymbol{\Pi}_n = \boldsymbol{Q}\left[\boldsymbol{R}_1,\, \boldsymbol{R}_2\right],$$

as computed via the Golub-Businger algorithm [227] where $\boldsymbol{\Pi}_n \in \{0,1\}^{n\times n}$ is a permutation matrix, $\boldsymbol{Q}$ is a unitary matrix, $\boldsymbol{R}_1$ is an upper triangular matrix, and $\boldsymbol{R}_2 \in \mathbb{C}^{m\times(n-m)}$.

The ordinary QR factorization is proceeded on $\boldsymbol{X}$ from the first column to the last column in order, whereas the order of the CPQR factorization is indicated by $\mathbf{\Pi}_n$. We refer to [227] for more details on the CPQR factorization.

Definition 6.3 (Projection onto $\mathcal{H}$ in (1.1))**.** *For an arbitrary matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, we define*

$$\mathcal{P}_{\mathcal{H}}(\boldsymbol{X}) := \underset{\boldsymbol{H}\in\mathcal{H}}{\arg\min}\, \|\boldsymbol{H}-\boldsymbol{X}\|_{\mathrm{F}} = \underset{\boldsymbol{H}\in\mathcal{H}}{\arg\max} \langle \boldsymbol{H}, \boldsymbol{X}\rangle$$

as the projection of $\boldsymbol{X}$ onto $\mathcal{H}$.

The projection aims to find the cluster structure that has the largest overall score given by $\boldsymbol{X}$. It is shown in [228] that projection onto $\mathcal{H}$ is equivalent to a *minimum-cost assignment problem* (MCAP) and can be efficiently solved by the "incremental algorithm" for MCAP [229, Section 3] with $\mathcal{O}(n^2 m \log m)$ computational complexity. The uniqueness condition of the projection $\mathcal{P}_{\mathcal{H}}(\boldsymbol{X})$ can be found in the proof of [229, Theorem 2.1] and [230, Theorem 2]. If the solution is not unique, the "incremental algorithm" for MCAP [229, Section 3] will generates a feasible projection randomly.

6.2 Problem Formulation

In this section, we formally define the probabilistic model, SBM-Ph, studied in this chapter. We first consider discrete phase angles and formulate the corresponding MLE problem which exhibits a *multi-frequency* structure. Then, we extend the problem to continuous phase angles and formulate a truncated MLE problem.

6.2.1 Stochastic Block Model with Discrete Relative Phase Angles

SBM-Ph is considered in a network with N nodes and $M \geq 2$ underlying clusters of equal size $s = {}^{N}\!/_{M}$. We assume each node $i \in [N]$ falls into one of M underlying clusters with the assignment $\mathcal{M}^*(i) \in [M]$ and is associated with an unknown phase angle $\theta_i^* \in \Omega$, where $\Omega := \{0, \dots, (2K_{\max}+1)\Delta\}$ is a discretization of $[0, 2\pi)$ with $\Delta = {}^{2\pi}/_{(2K_{\max}+1)}$. We use $\mathcal{S}_m^*$ to denote the set of nodes belonging to the mth cluster for all $m \in [M]$.

SBM-Ph generates a random graph $\mathcal{G} = ([N], \mathcal{E})$ with the node set $[N]$ and the edge set $\mathcal{E} \subseteq [N] \times [N]$. Each pair of nodes (i, j) are connected independently with probability p if i and j belong to the same cluster or equivalently, $\mathcal{M}^*(i) = \mathcal{M}^*(j)$. Otherwise, i and j are connected independently with probability q if $\mathcal{M}^*(i) \neq \mathcal{M}^*(j)$. Meanwhile, a relative phase angle $\theta_{ij} \in \Omega$ is observed on each edge $(i, j) \in \mathcal{E}$. When $\mathcal{M}^*(i) = \mathcal{M}^*(j)$, we obtain $\theta_{ij} := (\theta_i^* - \theta_j^*) \mod 2\pi$. Otherwise, we observe $\theta_{ij} := u_{ij} \sim \text{Unif}(\Omega)$, which is drawn uniformly at random from Ω.

Our observation model can be represented by the *observation matrix* $\boldsymbol{A} \in \mathbb{C}^{N\times N}$, which is a Hermitian matrix whose (i, j)th entry for any $i < j$ satisfies,

$$\boldsymbol{A}_{ij} = \begin{cases} e^{\iota(\theta_i^* - \theta_j^*)}, & \text{with prob } p \text{ if } \mathcal{M}^*(i) = \mathcal{M}^*(j), \\ e^{\iota u_{ij}}, & \text{with prob } q \text{ if } \mathcal{M}^*(i) \neq \mathcal{M}^*(j), \\ 0, & \text{o.w.,} \end{cases} \tag{6.1}$$

where $\boldsymbol{A}_{ji} = \overline{\boldsymbol{A}_{ij}}$. We also set the diagonal entry $\boldsymbol{A}_{ii} = 0, \forall i \in [N]$. Notice that a realization generated by the above observation matrix (6.1) is a noisy version of the *clean observation matrix* $\boldsymbol{A}^{\text{clean}} \in \mathbb{C}^{N\times N}$ whose (i, j)th entry satisfies,

$$\boldsymbol{A}_{ij}^{\text{clean}} = \begin{cases} e^{\iota(\theta_i^* - \theta_j^*)}, & \text{if } \mathcal{M}^*(i) = \mathcal{M}^*(j), \\ 0, & \text{otherwise.} \end{cases} \tag{6.2}$$

Specially, $\boldsymbol{A}$ is equal to $\boldsymbol{A}^{\text{clean}}$ when $p = 1$ and $q = 0$.

Remark 6.1. *Unlike the observation matrix (or adjacency matrix) $\boldsymbol{A}_{SBM}$ in SBM [93, 92, 83, 103] with only $\{0, 1\}$-valued entries, $\boldsymbol{A}$ in* (6.1) *extends to incorporating the relative phase angles θ_{ij} into edges. On the other hand, while entries of the observation matrix $\boldsymbol{A}_{Ph}$ in the phase synchronization problem [84, 121, 122] encode the the pairwise transformation information, they do not have the underlying M-cluster structure.*

6.2.2 MLE with Multi-Frequency Nature

Based on the observation matrix $\boldsymbol{A}$, we detail the MLE formulation for recovering the cluster structure and phase angles in this section. Given parameters, phase angles associated with nodes $\{\theta_i \in \Omega\}_{i=1}^N$ and the cluster structure $\{\mathcal{S}_m\}_{m=1}^M$ of equal size s, the probability model of observing $\boldsymbol{A}_{ij}$ between node pair (i, j) is

$$\begin{aligned}
&\mathbb{P}\left(\boldsymbol{A}_{ij} \,\middle|\, \{\theta_i \in \Omega\}_{i=1}^N, \{\mathcal{S}_m\}_{m=1}^M\right) \\
&= \begin{cases} p, & \text{if } \boldsymbol{A}_{ij} = e^{\iota(\theta_i - \theta_j)} \text{ and } \mathcal{M}(i) = \mathcal{M}(j), \\ 0, & \text{if } \boldsymbol{A}_{ij} \neq e^{\iota(\theta_i - \theta_j)} \text{ and } \mathcal{M}(i) = \mathcal{M}(j), \\ 1-p, & \text{if } \boldsymbol{A}_{ij} = 0 \text{ and } \mathcal{M}(i) = \mathcal{M}(j), \\ {}^{q}\!/\!{}_{K}, & \text{if } \boldsymbol{A}_{ij} = e^{\iota u_{ij}} \text{ and } \mathcal{M}(i) \neq \mathcal{M}(j), \\ 1-q, & \text{if } \boldsymbol{A}_{ij} = 0 \text{ and } \mathcal{M}(i) \neq \mathcal{M}(j), \end{cases}
\end{aligned}$$

where $\mathcal{M}(\cdot)$ is the assignment function corresponding to the cluster structure $\{\mathcal{S}_m\}_{m=1}^M$, and $K = 2K_{\max} + 1$. The likelihood function given observations on the edge set $\mathcal{E}$ is

$$\begin{aligned}
&\mathbb{P}\left(\{\boldsymbol{A}_{ij}\}_{(i,j)\in\mathcal{E}} \,\middle|\, \{\theta_i \in \Omega\}_{i=1}^N, \{\mathcal{S}_m\}_{m=1}^M\right) \\
&= \prod_{\substack{\mathcal{M}(i)=\mathcal{M}(j)\\(i,j)\in\mathcal{E}}} p^{\mathbb{I}\{\boldsymbol{A}_{ij} = e^{\iota(\theta_i - \theta_j)}\}} \prod_{\substack{\mathcal{M}(i)=\mathcal{M}(j)\\(i,j)\in\mathcal{E}}} 0^{\mathbb{I}\{\boldsymbol{A}_{ij} \neq e^{\iota(\theta_i - \theta_j)}\}} \prod_{\substack{\mathcal{M}(i)\neq\mathcal{M}(j)\\(i,j)\in\mathcal{E}}} {}^{q}\!/\!{}_{K},
\end{aligned} \tag{6.3}$$

due to the independence among edges within $\mathcal{E}$. Notice that maximizing the likelihood function (6.3) is equal to maximizing the following log-likelihood function

$$\begin{aligned}
&\log \mathbb{P}\left(\{\boldsymbol{A}_{ij}\}_{(i,j)\in\mathcal{E}} \,\middle|\, \{\theta_i \in \Omega\}_{i=1}^N, \{\mathcal{S}_m\}_{m=1}^M\right) = \\
&\sum_{\substack{\mathcal{M}(i)=\mathcal{M}(j)\\(i,j)\in\mathcal{E}}} \mathbb{I}\{\boldsymbol{A}_{ij} = e^{\iota(\theta_i - \theta_j)}\} \log p + \sum_{\substack{\mathcal{M}(i)=\mathcal{M}(j)\\(i,j)\in\mathcal{E}}} \mathbb{I}\{\boldsymbol{A}_{ij} \neq e^{\iota(\theta_i - \theta_j)}\} \log 0 \\
&+ \sum_{\substack{\mathcal{M}(i)\neq\mathcal{M}(j)\\(i,j)\in\mathcal{E}}} \log {}^{q}\!/\!{}_{K}.
\end{aligned} \tag{6.4}$$

Given $0 < {}^{q}\!/\!_{K} < p$, maximizing (6.4) is equivalent to

$$\max_{\substack{\{\theta_i \in \Omega\}_{i=1}^N \\ \{\mathcal{S}_m\}_{m=1}^M}} \sum_{\substack{\mathcal{M}(i)=\mathcal{M}(j) \\ (i,j)\in\mathcal{E}}} \mathbb{I}\{\theta_{ij} = [(\theta_i - \theta_j) \mod 2\pi]\}, \tag{6.5}$$

by assuming $0 \log 0 = 0$ in (6.4). By taking the FFT w.r.t. the support Ω of $((\theta_i - \theta_j) \mod 2\pi)$s and inverse FFT (IFFT) back, (6.5) is equivalent to

$$\max_{\substack{\{\theta_i \in \Omega\}_{i=1}^N \\ \{\mathcal{S}_m\}_{m=1}^M}} \sum_{k=-K_{\max}}^{K_{\max}} \sum_{m=1}^{M} \sum_{i,j\in\mathcal{S}_m} \left\langle \boldsymbol{A}_{ij}^{(k)}, e^{\iota k(\theta_i - \theta_j)} \right\rangle, \tag{6.6}$$

where $\boldsymbol{A}^{(k)}$ is the kth entry-wise power of $\boldsymbol{A}$ with $\boldsymbol{A}_{ij}^{(k)} = e^{\iota k \theta_{ij}}$.

As indicated by (6.6), the MLE exhibits a *multi-frequency* nature, where the kth frequency component is $\sum_{m=1}^{M} \sum_{i,j\in\mathcal{S}_m} \langle \boldsymbol{A}_{ij}^{(1)}, e^{\iota(\theta_i - \theta_j)} \rangle$ in (6.6). Although the following program using the first frequency component

$$\max_{\substack{\{\theta_i \in \Omega\}_{i=1}^N \\ \{\mathcal{S}_m\}_{m=1}^M}} \sum_{m=1}^{M} \sum_{i,j\in\mathcal{S}_m} \left\langle \boldsymbol{A}_{ij}^{(1)}, e^{\iota(\theta_i - \theta_j)} \right\rangle, \tag{6.7}$$

is a reasonable formulation for the joint estimation problem as suggested by [123, 1, 2], it is indeed not a MLE formulation. One can show that (6.7) is equivalent to

$$\max_{\substack{\{\theta_i \in \Omega\}_{i=1}^N \\ \{\mathcal{S}_m\}_{m=1}^M}} \sum_{\substack{\mathcal{M}(i)=\mathcal{M}(j) \\ (i,j)\in\mathcal{E}}} \cos(\theta_{ij} - (\theta_i - \theta_j)),$$

which is not the MLE (6.5) of the joint estimation problem.

To proceed, we perform a change of optimization variables for (6.6). By defining a unitary matrix $\boldsymbol{V} \in \mathbb{C}^{N\times M}$ whose (i,m)th entry satisfies

$$\boldsymbol{V}_{im} := \begin{cases} \frac{1}{\sqrt{s}} e^{\iota\theta_i}, & \text{if } i \in \mathcal{S}_m (\text{or } \mathcal{M}(i) = m), \\ 0, & \text{otherwise}, \end{cases} \tag{6.8}$$

the cluster structure $\{\mathcal{S}_m\}_{m=1}^M$ and the associated phase angles $\{\theta_i \in \Omega\}_{i=1}^N$ are encoded into one simple unitary matrix $\boldsymbol{V}$. Then, the optimization program (6.6) can be reformulated as

$$\begin{aligned} \max_{\boldsymbol{V} \in \mathbb{C}^{N \times M}} \quad & \sum_{k=-K_{\max}}^{K_{\max}} \left\langle \boldsymbol{A}^{(k)}, \boldsymbol{V}^{(k)} \left(\boldsymbol{V}^{(k)}\right)^{\mathsf{H}} \right\rangle \\ \text{s.t.} \quad & \boldsymbol{V} \text{ satisfies the form (6.8)}, \end{aligned} \tag{6.9}$$

where each $\boldsymbol{V}^{(k)}$ is generated by $\boldsymbol{V}$ through the entry-wise power that satisfies

$$\boldsymbol{V}_{im}^{(k)} := \begin{cases} \frac{1}{\sqrt{s}} e^{\iota k \theta_i}, & \text{if } i \in \mathcal{S}_m (\text{or } \mathcal{M}(i) = m), \\ 0, & \text{otherwise.} \end{cases} \tag{6.10}$$

The optimization program (6.9) is non-convex and is thus computationally intractable to be solved exactly. Although one can try SDP based approaches similar to [123], it is not guaranteed to obtain exact solutions to the MLE, let alone the high computational complexity when N and $K_{\max}$ are large. Therefore, we propose a spectral method based on the MF-CPQR factorization and an iterative MF-GPM in Section 6.3 and Section 6.4, respectively.

6.2.3 Extension to Continuous Phase Angles: A Truncated MLE

We consider the joint estimation problem on a discretization of $[0, 2\pi)$ in Section 6.2.1, and then derive the MLE formulation in Section 6.2.2. Now, we turn to the joint estimation problem with continuous phase angles in $[0, 2\pi)$ $(\theta_i \in [0, 2\pi), \forall i \in [N])$.

Following the similar steps as (6.3), (6.4), (6.5), the MLE formulation is

$$\max_{\substack{\{\theta_i \in [0,2\pi)\}_{i=1}^N \\ \{\mathcal{S}_m\}_{m=1}^M}} \sum_{\substack{\mathcal{M}(i)=\mathcal{M}(j) \\ (i,j) \in \mathcal{E}}} \mathbb{I}([(\theta_i - \theta_j) \mod 2\pi] = \theta_{ij}). \tag{6.11}$$

The MLE formulation (6.11) is essentially equal to counting the times that $\delta([(\theta_i - \theta_j) \mod 2\pi] = \theta_{ij}) = \infty$, where $\delta(\cdot)$ is the Dirac delta function. We can express the Dirac delta function with its Fourier series expansion,

$$\begin{aligned}\delta([(\theta_i - \theta_j) \mod 2\pi] = \theta_{ij}) &= \sum_{k=-\infty}^{+\infty} e^{\iota k(\theta_i - \theta_j)} e^{-\iota k \theta_{ij}} \\ &\approx \sum_{k=-K_{\max}}^{K_{\max}} e^{\iota k(\theta_i - \theta_j)} e^{-\iota k \theta_{ij}}.\end{aligned} \tag{6.12}$$

The straightforward truncation in (6.12) corresponds to approximating the Dirac delta with the *Dirichlet kernel*. By this truncation, the problem in (6.11) is converted to

$$\max_{\substack{\{\theta_i \in [0,2\pi)\}_{i=1}^N \\ \{\mathcal{S}_m\}_{m=1}^M}} \sum_{k=-K_{\max}}^{K_{\max}} \sum_{m=1}^{M} \sum_{i,j \in \mathcal{S}_m} \left\langle \boldsymbol{A}_{ij}^{(k)}, e^{\iota k(\theta_i - \theta_j)} \right\rangle. \tag{6.13}$$

The optimization program (6.13) is a truncated MLE of the joint estimation problem with continuous phase angles of (6.11).

As one can observe from (6.6) and (6.13), the only difference is that $\theta_i \in \Omega$ is discrete in (6.6), and $\theta_i \in [0, 2\pi)$ is continuous in (6.13). Algorithms in Section 6.3 and 6.4 can also be directly applied to the joint estimation problem with continuous phase angles after simple modification. Due to the similarity between the joint estimation problem and its continuous extension, we will only focus on the joint estimation problem on Ω (despite numerical experiments) in remaining parts of this chapter for brevity.

6.3 Spectral Method Based on the MF-CPQR Factorization

In this section, we propose a spectral method based on the novel MF-CPQR factorization for the joint estimation problem. We start with introducing main steps and motivations of Algorithm 6.1 in Section 6.3.1. Section 6.3.2 states the novel algorithm, the MF-CPQR factorization, designed for our spectral method, together with the difference between the MF-CPQR factorization and the CPQR factorization. In Section 6.3.3, we discuss the computational complexity of our proposed algorithm in detail.

Our spectral method based on the MF-CPQR factorization is inspired by

Algorithm 6.1: The spectral method based on the MF-CPQR factorization

```
Input: The observation matrix $\boldsymbol{A}$, and the number of clusters $M$.
(Eigendecomposition) For $k = -K_{\max}, \dots, K_{\max}$, compute the top $M$
    eigenvectors $\boldsymbol{\Phi}^{(k)} \in \mathbb{C}^{N \times M}$ of $\boldsymbol{A}^{(k)}$ such that $\left(\boldsymbol{\Phi}^{(k)}\right)^{\mathsf{H}} \boldsymbol{\Phi}^{(k)} = \boldsymbol{I}_M$
(MF-CPQR factorization) Compute the multi-frequency column-pivoted QR
    factorization (detailed in Algorithm 6.2) of $\left\{\left(\boldsymbol{\Phi}^{(k)}\right)^{\top}\right\}_{k=-K_{\max}}^{K_{\max}}$, which yields
        $$\left(\boldsymbol{\Phi}^{(k)}\right)^{\top} \boldsymbol{\Pi}_N = \boldsymbol{Q}^{(k)} \boldsymbol{R}^{(k)} \Rightarrow \left(\boldsymbol{\Phi}^{(k)}\right)^{\top} = \boldsymbol{Q}^{(k)} \boldsymbol{R}^{(k)} \boldsymbol{\Pi}_N^{\top} \qquad (6.14)$$
    Update $\boldsymbol{R}^{(k)} \leftarrow \boldsymbol{R}^{(k)} \boldsymbol{\Pi}_N^{\top}, \forall k = -K_{\max}, \dots, K_{\max}$
(Recovery of the cluster structure and the phase angles) For each node $i \in [N]$,
    assign its cluster as
        $$\hat{\mathcal{M}}(i) \leftarrow \arg\max_{m \in [M]} \left\{ \max_{\theta_i \in \Omega} \sum_{k=-K_{\max}}^{K_{\max}} \left\langle e^{\iota k \theta_i}, \boldsymbol{R}_{mi}^{(k)} \right\rangle \right\} \qquad (6.15)$$
    Then estimate the phase angle given the recovered cluster assignment $\hat{\mathcal{M}}(i)$
        $$\hat{\theta}_i \leftarrow \arg\max_{\theta_i \in \Omega} \sum_{k=-K_{\max}}^{K_{\max}} \left\langle e^{\iota k \theta_i}, \boldsymbol{R}_{\hat{\mathcal{M}}(i)i}^{(k)} \right\rangle \qquad (6.16)$$
Output: Estimated cluster structure $\{\hat{\mathcal{M}}(i)\}_{i=1}^{N}$ and estimated phase angles
        $\{\hat{\theta}_i\}_{i=1}^{N}$
```

the CPQR-type algorithms [231, 1] together with the *multi-frequency* nature of the MLE formulation (6.9). Similar to the CPQR-type algorithms, Algorithm 6.1 is deterministic and free of any initialization. Meanwhile, in terms of computational complexity, Algorithm 6.1 scales linearly w.r.t. the number of edges $|\mathcal{E}|$ and near-linearly w.r.t. $K_{\max}$.

6.3.1 Motivations

Algorithm 6.1 consists of three steps: i) Eigendecomposition of $\boldsymbol{A}^{(k)}$, ii) MF-CPQR factorization, and iii) Recovery of the cluster structure and phase angles. It first computes matrices $\{\boldsymbol{\Phi}^{(k)}\}_{k=-K_{\max}}^{K_{\max}}$ that contain the top M eigenvectors of each $\boldsymbol{A}^{(k)}$ via eigendecomposition. Secondly, matrices $\{\boldsymbol{R}^{(k)}\}_{k=-K_{\max}}^{K_{\max}}$ are obtained through the MF-CPQR factorization which is detailed in Algorithm 6.2. The last step is recovering the cluster structure and associated phase angles based on $\{\boldsymbol{R}^{(k)}\}_{k=-K_{\max}}^{K_{\max}}$ via (6.15) and (6.16).

In terms of motivations for Algorithm 6.1, we start from the MLE formulation (6.9). We first relax (6.9) by replacing the constraints in (6.8) with $\boldsymbol{V}^{\mathsf{H}}\boldsymbol{V} = \boldsymbol{I}_M$,

$$\begin{aligned} \boldsymbol{\Phi} = &\operatorname*{arg\,max}_{\boldsymbol{V}\in\mathbb{C}^{N\times M}} \sum_{k=-K_{\max}}^{K_{\max}} \left\langle \boldsymbol{A}^{(k)}, \boldsymbol{V}^{(k)} \left(\boldsymbol{V}^{(k)}\right)^{\mathsf{H}} \right\rangle \\ &\text{s.t. } \boldsymbol{V}^{\mathsf{H}}\boldsymbol{V} = \boldsymbol{I}_M, \end{aligned} \tag{6.17}$$

by noticing that $\boldsymbol{V}$ in (6.8) forms an orthonormal basis. The optimization problem in (6.17) is still non-convex, and there is no simple spectral method that can directly solve the problem. One approach is to relax the dependency of $\boldsymbol{V}^{(k)}$ among different frequencies and split (6.17) into different frequencies, and that is, for $k = -K_{\max}, \dots, K_{\max}$, we have

$$\begin{aligned} \boldsymbol{\Phi}^{(k)} = &\operatorname*{arg\,max}_{\boldsymbol{V}^{(k)}\in\mathbb{C}^{N\times M}} \left\langle \boldsymbol{A}^{(k)}, \boldsymbol{V}^{(k)} \left(\boldsymbol{V}^{(k)}\right)^{\mathsf{H}} \right\rangle \\ &\text{s.t. } \left(\boldsymbol{V}^{(k)}\right)^{\mathsf{H}}\boldsymbol{V}^{(k)} = \boldsymbol{I}_M. \end{aligned} \tag{6.18}$$

The optimizer of (6.18) is the matrix that contains the top M eigenvectors of $\boldsymbol{A}^{(k)}$ denoted by $\boldsymbol{\Phi}^{(k)} \in \mathbb{C}^{N\times M}$. This accounts for step 1 (eigendecomposition) in Algorithm 6.1.

In fact, one can infer the cluster structure from $\{\boldsymbol{\Phi}^{(k)}\}_{k=-K_{\max}}^{K_{\max}}$. To see this, for $k = -K_{\max}, \dots, K_{\max}$, we split $\boldsymbol{A}^{(k)}$ into deterministic and random parts:

$$\boldsymbol{A}^{(k)} = \mathbb{E}[\boldsymbol{A}^{(k)}] + (\boldsymbol{A}^{(k)} - \mathbb{E}[\boldsymbol{A}^{(k)}]) = \mathbb{E}[\boldsymbol{A}^{(k)}] + \boldsymbol{\Delta}^{(k)}, \tag{6.19}$$

where $\mathbb{E}[\boldsymbol{A}^{(k)}] = p\boldsymbol{A}_{\text{clean}}^{(k)}$ with $\boldsymbol{A}_{\text{clean}}^{(k)}$ being the entry-wise kth power of $\boldsymbol{A}_{\text{clean}}$ (6.2), and the residual $\boldsymbol{\Delta}^{(k)}$ is a random perturbation with $\mathbb{E}[\boldsymbol{\Delta}^{(k)}] = \boldsymbol{0}$. Obviously, each $\mathbb{E}[\boldsymbol{A}^{(k)}]$ is a low rank matrix that satisfies the following eigendecomposition:

$$\begin{aligned} &\mathbb{E}[\boldsymbol{A}^{(k)}] = ps \sum_{m=1}^{M} \boldsymbol{\Psi}_{\cdot,m}^{(k)} \left(\boldsymbol{\Psi}_{\cdot,m}^{(k)}\right)^{\mathsf{H}}, \\ &\text{with} \quad \boldsymbol{\Psi}_{im}^{(k)} := \begin{cases} \frac{1}{\sqrt{s}} e^{\iota k \theta_i^*}, & \text{if } i \in \mathcal{S}_m^*, \\ 0, & \text{otherwise}, \end{cases} \end{aligned}$$

where $\boldsymbol{\Psi}^{(k)} \in \mathbb{C}^{N\times M}$ is a matrix defined in a similar manner as $\boldsymbol{V}^{(k)}$ in (6.10), and satisfies $\left(\boldsymbol{\Psi}^{(k)}\right)^{\mathsf{H}}\boldsymbol{\Psi}^{(k)} = \boldsymbol{I}_M$. Then, for $k = -K_{\max}, \dots, K_{\max}$ (except for

$k = 0$), the non-zero entry in each row of $\mathbf{\Psi}^{(k)}$ indicates the underlying cluster assignment $\mathcal{M}^*(i)$ and the exact phase angle θ_i^* of node i.

Therefore, to recover the cluster structure and associated phase angles, it suffices to extract $\{\mathbf{\Psi}^{(k)}\}_{k=-K_{\max}}^{K_{\max}}$ from $\{\mathbf{\Phi}^{(k)}\}_{k=-K_{\max}}^{K_{\max}}$. For the ease of illustration, we first consider the case when $p = 1$ and $q = 0$. This indicates, for $k = -K_{\max}, \dots, K_{\max}$, $\boldsymbol{A}^{(k)} = \boldsymbol{A}^{(k)}_{\text{clean}}$, $\mathbf{\Delta}^{(k)} = \mathbf{0}$, and $\mathbf{\Phi}^{(k)} = \mathbf{\Psi}^{(k)}\boldsymbol{O}^{(k)}$, where $\boldsymbol{O}^{(k)} \in \mathbb{C}^{M\times M}$ is some unitary matrix. However, $\{\boldsymbol{O}^{(k)}\}_{k=-K_{\max}}^{K_{\max}}$ are unknown and even not synchronized among all frequencies. To address this issue, the MF-CPQR factorization is introduced. Here, we assume that the first s nodes are from the cluster $\mathcal{S}_1^*$, the following s nodes are from $\mathcal{S}_2^*$, and so on. Applying the MF-CPQR factorization (step 2) in Algorithm 6.1 yields (assume $\mathbf{\Pi}_N = \boldsymbol{I}_N$)

$$
\begin{aligned}
&\left(\mathbf{\Phi}^{(k)}\right)^\top = \left(\boldsymbol{O}^{(k)}\right)^\top \left(\mathbf{\Psi}^{(k)}\right)^\top = \frac{1}{\sqrt{s}} \left(\boldsymbol{O}^{(k)}\right)^\top \times \\
&\begin{bmatrix} e^{\iota k\theta_1^*} & \cdots & e^{\iota k\theta_s^*} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \cdots & e^{\iota k\theta_{N-s+1}^*} & \cdots & e^{\iota k\theta_N^*} \end{bmatrix} \\
&= \underbrace{\left(\boldsymbol{O}^{(k)}\right)^\top \begin{bmatrix} e^{\iota k\theta_1^*} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{\iota k\theta_{N-s+1}^*} \end{bmatrix}}_{=:\boldsymbol{Q}^{(k)}} \times \\
&\underbrace{\begin{bmatrix} 1 & \cdots & e^{\iota k(\theta_s^*-\theta_1^*)} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 & \cdots & e^{\iota k(\theta_N^*-\theta_{N-s+1}^*)} \end{bmatrix}}_{=:\boldsymbol{R}^{(k)}} \\
&= \boldsymbol{Q}^{(k)}\boldsymbol{R}^{(k)},
\end{aligned}
\tag{6.20}
$$

for $k = -K_{\max}, \dots, K_{\max}$. Therefore, each $\boldsymbol{Q}^{(k)} \in \mathbb{C}^{M\times M}$ is a unitary matrix that includes the unknown unitary matrix $\boldsymbol{O}^{(k)}$, and each $\boldsymbol{R}^{(k)} \in \mathbb{C}^{M\times N}$ is a matrix that excludes $\boldsymbol{O}^{(k)}$. More significantly, $\{\boldsymbol{R}^{(k)}\}_{k=-K_{\max}}^{K_{\max}}$ contains all the information needed to recover the cluster structure and associated phase angles.

To recover the cluster structure, the CPQR-type algorithm [1] only uses $\boldsymbol{R}^{(1)}$. By noticing that for each node i, the ith column of $\boldsymbol{R}^{(1)}$ (e.g., $\boldsymbol{R}^{(1)}_{:,i}$)

is sparse (its mth entry $\boldsymbol{R}^{(1)}_{mi}$ is nonzero if and only if $m = \mathcal{M}^*(i)$), one can determine the cluster assignment of node i by the position of the nonzero entry. Meanwhile, the associated phase angle can also be determined by obtaining the phase angle from the nonzero entry (up to some global phase transition in the same cluster). When the observation $\boldsymbol{A}$ is noisy, the CPQR-type algorithm recovers the cluster structure and associated phase angle of node i by the position of the entry with the largest amplitude. The following Theorem 6.1 proves as long as the perturbation to $\mathbb{E}[\boldsymbol{A}^{(k)}]$ is less than a certain threshold, $\boldsymbol{\Phi}^{(k)}$ is still close to $\boldsymbol{\Psi}^{(k)}\boldsymbol{O}^{(k)}$, for $k = -K_{\max}, \ldots, K_{\max}$ (except for $k = 0$).

Theorem 6.1 (Row-wise error bound, adapted from [1])**.** *Given a network with N nodes and $M = 2$ underlying clusters, for a sufficiently large N, we suppose*

$$\eta := \frac{\sqrt{(p(1-p)+q)\log N}}{p\sqrt{N}} \leq c_0$$

for some small constant c_0. Consequently, with probability at least $1 - \mathcal{O}(N^{-1})$,

$$\max_{i\in[N]} \quad \left\| \boldsymbol{\Phi}^{(k)}_{i,\cdot} - \boldsymbol{\Psi}^{(k)}_{i,\cdot}\boldsymbol{O}^{(k)} \right\|_2 \lesssim \frac{\eta}{\sqrt{N}},$$

where $\boldsymbol{O}^{(k)} = \mathcal{P}((\boldsymbol{\Psi}^{(k)})^{\mathsf{H}}\boldsymbol{\Phi}^{(k)})$.

Theorem 6.1 guarantees that i) amplitudes of other entries are less than the entry indicating the true cluster structure with high probability, ii) the phase angle information is preserved with high fidelity. Theorem 6.1 can be proven by following the same routines as [1] by replacing the orthogonal group element $\boldsymbol{O}_i$ with the $U(1)$ group element (e.g., $e^{\iota\theta_i}$). The reason why Theorem 6.1 holds for $k = -K_{\max}, \ldots, K_{\max}$ (despite $k = 0$) is due to statistics of random perturbations $\{\boldsymbol{\Delta}^{(k)}\}_k$ in (6.19) do not change among different frequencies. This is because the noise models of $\boldsymbol{A}^{(k)}$ and $\boldsymbol{A}$ are the same. More specifically, the noisy entry $e^{\iota u_{ij}} : u_{ij} \sim \text{Unif}(\Omega)$ in (6.1) has the same statistics as $e^{\iota k u_{ij}}$ in $\boldsymbol{A}^{(k)}$ due to the fact that ku_{ij} still yields the distribution $\text{Unif}(\Omega)$.

Note that the CPQR-type algorithm in [1] is not developed from the MLE formulation (6.9) of the joint estimation problem and thus does not capture the *multi-frequency* nature. In this chapter, we leverage $\{\boldsymbol{R}^{(k)}\}_{k=-K_{\max}}^{K_{\max}}$ that contain information about the cluster structure and associated phase angles

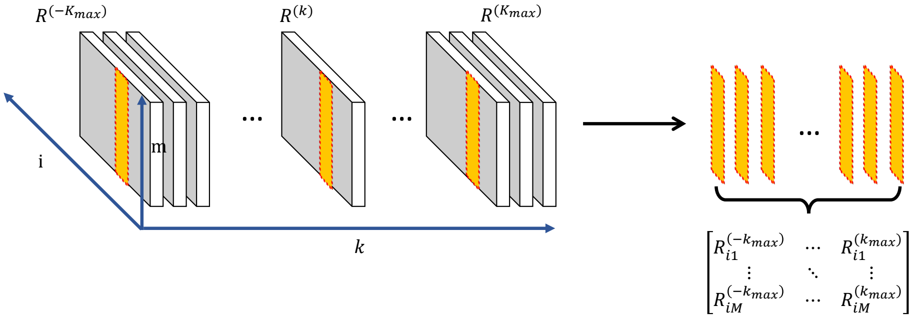

$R^{(-K_{max})}$
$R^{(k)}$
$R^{(K_{max})}$
i
m
k
...
$\begin{bmatrix} R_{i1}^{(-k_{max})} & \cdots & R_{i1}^{(k_{max})} \\ \vdots & \ddots & \vdots \\ R_{iM}^{(-k_{max})} & \cdots & R_{iM}^{(k_{max})} \end{bmatrix}$

Figure 6.1: Illustration of step 3 in Algorithm 6.1. For each $i \in [N]$, all the ith columns in $\left\{\boldsymbol{R}^{(k)}\right\}_{k=-K_{\max}}^{K_{\max}}$ are extracted to estimate the cluster assignment and phase angle following (6.15) and (6.16).

across multiple frequencies (step 3). Specifically, we first consider the same case as that in (6.20) for intuition. As illustrated in Figure 6.1, the matrix concatenated by the ith ($i \leq s$) columns across all frequencies is

$$\begin{bmatrix} \boldsymbol{R}_{i1}^{(-K_{\max})} & \cdots & \boldsymbol{R}_{i1}^{(k)} & \cdots & \boldsymbol{R}_{i1}^{(K_{\max})} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{R}_{iM}^{(-K_{\max})} & \cdots & \boldsymbol{R}_{iM}^{(k)} & \cdots & \boldsymbol{R}_{iM}^{(K_{\max})} \end{bmatrix} = \frac{1}{\sqrt{s}} \times$$
$$\begin{bmatrix} e^{-\iota K_{\max}(\theta_i^* - \theta_1^*)} & \cdots & e^{\iota k(\theta_i^* - \theta_1^*)} & \cdots & e^{\iota K_{\max}(\theta_i^* - \theta_1^*)} \\ 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \end{bmatrix}.$$

The cluster assignment of i can be acquired by finding the non-sparse row of the above matrix, and the phase angle can be determined by evaluating the non-sparse row (e.g., FFT). When the observation $\boldsymbol{A}$ is noisy, (6.15) and (6.16) are used to estimate the cluster structure and associated phase angles, which can be interpreted as checking the consistency or conducting majority vote among all frequencies. The performance is expected to be at least as good as the CPQR-type algorithm. This is because each $\boldsymbol{\Phi}^{(k)}$ has the same theoretical guarantee as the CPQR-type algorithm according to Theorem 6.1, and (6.15) (6.16) are just checking the consistency across all frequencies. In Section 6.5, we will show that our proposed spectral method based on the MF-CPQR factorization is capable of significantly outperforming the CPQR-type algorithm.

Algorithm 6.2: MF-CPQR factorization

```
Input: The set of eigenvectors $\left\{\left(\mathbf{\Phi}^{(k)}\right)\right\}_{k=-K_{\max}}^{K_{\max}}$
Init: $\boldsymbol{Q}^{(k)} \leftarrow \boldsymbol{I}_M$, $\boldsymbol{R}^{(k)} \leftarrow \left(\mathbf{\Phi}^{(k)}\right)^\top, \forall i = -K_{\max}, \ldots, K_{\max}$, and $\mathbf{\Pi}_N \leftarrow \boldsymbol{I}_N$
for $m = 1, 2, \ldots, M$ do
    /* Pivot selection */
    for $j = m, m+1, \ldots, N$ do
        Compute the residual $\rho_j \leftarrow \sum_{k=-K_{\max}}^{K_{\max}} \|\boldsymbol{R}^{(k)}_{m:,\, j}\|_2$
    end
    Determine the pivot $j^* \leftarrow \arg\max_{j=m,\ldots,N} \quad \rho_j$
    For both $\{\boldsymbol{R}^{(k)}\}_{k=1}^{K-1}$ and $\mathbf{\Pi}_N$, swap the $m$th column with the pivot ($j^*$th)
      column
    /* One step QR factorization for all frequencies */
    for $k = -K_{max}, \ldots, K_{max}$ do
        Apply one step QR factorization in Algorithm 6.3 on $\boldsymbol{R}^{(k)}_{m:,m:}$, and get
          $\widetilde{\boldsymbol{Q}}^{(k)}_{m:,m:}$ and $\widetilde{\boldsymbol{R}}^{(k)}_{m:,m:}$
        Update $\boldsymbol{Q}^{(k)}_m \leftarrow \begin{bmatrix} \boldsymbol{I}_{m-1} & \mathbf{0} \\ \mathbf{0} & \widetilde{\boldsymbol{Q}}^{(k)}_{m:,m:} \end{bmatrix}$
        Update $\boldsymbol{R}^{(k)}_{m:,m:} \leftarrow \widetilde{\boldsymbol{R}}^{(k)}_{m:,m:}$ and $\boldsymbol{Q}^{(k)} \leftarrow \boldsymbol{Q}^{(k)}\boldsymbol{Q}^{(k)}_m$
    end
end
Output: $\{\boldsymbol{Q}^{(k)}\}_{k=-K_{\max}}^{K_{\max}}$, $\{\boldsymbol{R}^{(k)}\}_{k=-K_{\max}}^{K_{\max}}$, and $\mathbf{\Pi}_N$
```

Besides, for the joint estimation problem with continuous phase angles, (6.15) and (6.16) will be modified as

$$\hat{\mathcal{M}}(i) \leftarrow \arg\max_{m \in [M]} \quad \left\{ \max_{\theta_i \in [0, 2\pi)} \quad \sum_{k=-K_{\max}}^{K_{\max}} \left\langle e^{\iota k \theta_i}, \boldsymbol{R}^{(k)}_{mi} \right\rangle \right\},$$

$$\hat{\theta}_i \leftarrow \arg\max_{\theta_i \in [0, 2\pi)} \quad \sum_{k=-K_{\max}}^{K_{\max}} \left\langle e^{\iota k \theta_i}, \boldsymbol{R}^{(k)}_{\hat{\mathcal{M}}(i) i} \right\rangle .$$

Solving the max problem over $[0, 2\pi)$ is infeasible in general. Instead, one can apply the zero-padding and FFT for an approximate solution with any desired precision. Specifically, in estimating the cluster assignment, by padding zeros to $[\boldsymbol{R}^{(-K_{\max})}_{mi}, \ldots, \boldsymbol{R}^{(k)}_{mi}, \ldots, \boldsymbol{R}^{(K_{\max})}_{mi}]$ as $[0, \ldots, 0, \boldsymbol{R}^{(-K_{\max})}_{mi}, \ldots, \boldsymbol{R}^{(k)}_{mi}, \ldots, \boldsymbol{R}^{(K_{\max})}_{mi}, 0, \ldots, 0]$, taking the FFT, and finding the entry with largest real part, $(\arg)\max_{\theta_i \in [0, 2\pi)} \sum_{k=-K_{\max}}^{K_{\max}} \left\langle e^{\iota k \theta_i}, \boldsymbol{R}^{(k)}_{mi} \right\rangle$ can be solved approximately, where the precision is determined by the number of padded zeros.

Algorithm 6.3: One step QR factorization using Householder transformation

```
Input: A matrix $\boldsymbol{X} \in \mathbb{C}^{n\times n}$
/* Householder transformation */
$\boldsymbol{r} \leftarrow \boldsymbol{X}_{\cdot,1}$
$\theta \leftarrow -e^{\iota\angle \boldsymbol{r}_1}\|\boldsymbol{r}\|$, where $\angle \boldsymbol{r}_1$ is the phase angle of $\boldsymbol{r}_1$
$\boldsymbol{u} \leftarrow \boldsymbol{r} - \theta\boldsymbol{e}$, where $\boldsymbol{e} = [1, 0, \ldots, 0]^\top$
$\boldsymbol{v} \leftarrow {}^{\boldsymbol{u}}/_{\|\boldsymbol{u}\|}$
$\boldsymbol{Q} \leftarrow \boldsymbol{I}_n - 2\boldsymbol{v}\boldsymbol{v}^{\mathsf{H}}$
$\boldsymbol{X} \leftarrow \boldsymbol{Q}\boldsymbol{X}$
$\boldsymbol{X}_{1,\cdot} \leftarrow e^{-\iota\angle \boldsymbol{X}_{11}} X_{1,\cdot}$
$\boldsymbol{Q}_{\cdot,1} \leftarrow e^{\iota\angle \boldsymbol{X}_{11}} \boldsymbol{Q}_{\cdot,1}$
Output: $\boldsymbol{Q}$ and $\boldsymbol{R}$.
```

6.3.2 MF-CPQR Factorization

As stated in Definition 6.2, the difference between the ordinary QR factorization and the CPQR factorization is selecting appropriate pivot ordering (encoded in $\boldsymbol{\Pi}_N$). The CPQR factorization attempts to find a subset of columns that are as most linearly independent as possible and are used to determine the basis. In this chapter, the CPQR factorization across multiple frequencies is developed to cope with the *multi-frequency* structure of the MLE formulation.

Definition 6.4 (Multi-frequency column-pivoted QR factorization)**.** *Let* $\boldsymbol{X}^{(k)} \in \mathbb{C}^{m\times n}$ *with* $m \le n$ *has rank* m *for* $k = -K_{max}, \ldots, K_{max}$. *The multi-frequency column-pivoted QR factorization of* $\boldsymbol{X}^{(k)}$ *is the factorization*

$$\boldsymbol{X}^{(k)}\boldsymbol{\Pi}_n = \boldsymbol{Q}^{(k)}\left[\boldsymbol{R}_1^{(k)},\, \boldsymbol{R}_2^{(k)}\right],$$

as computed via Algorithm 6.2 where $\boldsymbol{\Pi}_n \in \{0,1\}^{n\times n}$ *is a permutation matrix fixed for all* $k = -K_{max}, \ldots, K_{max}$, $\boldsymbol{Q}^{(k)}$ *is a unitary matrix,* $\boldsymbol{R}_1^{(k)}$ *is an upper triangular matrix, and* $\boldsymbol{R}_2^{(k)} \in \mathbb{C}^{m\times(n-m)}$.

It requires to i) obtain the same subset of columns among all frequencies that are as most linearly independent as possible, and ii) use the same pivot ordering (or $\boldsymbol{\Pi}_N$) among all frequencies. The former promotes the cluster structure estimation performance because each node i (other than the pivots) is assigned to a cluster mainly according to the similarities between the column i and the columns of pivots: the latter ensures the validity of (6.15) and (6.16).

The MF-CPQR factorization is detailed in Algorithm 6.2, where the Householder transform [226] (Algorithm 6.3) is adopted for a better numerical stability. Specifically, the novel MF-CPQR factorization is different from the ordinary CPQR [232, 225] in the pivot selection. The pivot is determined by finding the column with the largest summation of ℓ_2 norm of residuals over all frequencies (see line 3 in Algorithm 6.2).

Table 6.2: The computational complexity of Algorithm 6.1 in each step.

Steps	Computational Complexity
1. Eigendecomposition	$\mathcal{O}(K_{\max}\lvert\mathcal{E}\rvert)$
2. MF-CPQR factorization	$\mathcal{O}(K_{\max}N)$
3. Clustering by (6.15)	$\mathcal{O}(NK_{\max}\log K_{\max})$
4. Phase synchronization by (6.16)	$\mathcal{O}(N)$
Total complexity	$\mathcal{O}(K_{\max}(\lvert\mathcal{E}\rvert + N\log K_{\max}))$

6.3.3 Computational Complexity

In this section, the computational complexity of Algorithm 6.1 is summarized step by step in Table 6.2. Here, we suppose $M = \Theta(1)$. First, it consists of $\mathcal{O}(K_{\max})$ times of eigendecomposition for M eigenvectors, which is $\mathcal{O}(|\mathcal{E}|)$ per time if using Lanczos method [233]. For the MF-CPQR factorization, it consists of M times of column pivoting ($\mathcal{O}(NK_{\max})$ per time) and $MK_{\max}$ times of one step QR factorization ($\mathcal{O}(N)$ per step). In terms of recovering the cluster structure, we first compute MN times of FFT for length-$K_{\max}$ vectors ($\mathcal{O}(K_{\max}\log K_{\max})$ per vector) and then compute the maximums ($\mathcal{O}(NK_{\max}) + \mathcal{O}(N)$). Since the FFT of $\{\boldsymbol{R}^{(k)}\}_{k=-K_{\max}}^{K_{\max}}$ is already computed, it is only $\mathcal{O}(N)$ for synchronizing the phase angles. Overall, the computational cost is linear with the number of edges $|\mathcal{E}|$ and nearly linear in $K_{\max}$. When the network $\mathcal{G}$ is densely connected with $|\mathcal{E}| = \mathcal{O}(N^2)$, Algorithm 6.1 ends up with $\mathcal{O}(K_{\max}N^2)$ if $\log K_{\max} < N$. However, if $|\mathcal{E}| = o(N^2)$, the complexity of Algorithm 6.1 will be reduced. For instance, in the case when $|\mathcal{E}| = \mathcal{O}(N\log N)$ or $|\mathcal{E}| = \mathcal{O}(N)$, which is very common as shown in [234], the complexity of Algorithm 6.1 will be $\mathcal{O}(K_{\max}N\max\{\log N, \log K_{\max}\})$ or $\mathcal{O}(K_{\max}N\log K_{\max}\})$, respectively.

Algorithm 6.4: Iterative multi-frequency generalized power method

```
Input: The observation matrix $\boldsymbol{A}$, the initialization $\{\mathcal{S}_m\}_{m=1}^M$ and $\{\theta_i \in \Omega\}_{i=1}^N$,
       and the number of iterations $T$
Construct $\{\widehat{\boldsymbol{V}}^{(k),0}\}_{k=-K_{\max}}^{K_{\max}}$ using $\{\mathcal{S}_m\}_{m=1}^M$ and $\{\theta_i \in \Omega\}_{i=1}^N$ according to (6.10)
for $t = 0, 1, \ldots, T-1$ do
    /* Matrix multiplication */
    For $k = -K_{\max}, \ldots, K_{\max}$, compute the matrix multiplication
      $\widehat{\boldsymbol{V}}^{(k),t+1} \leftarrow \boldsymbol{A}^{(k)} \widehat{\boldsymbol{V}}^{(k),t}$
    /* Combine information across multiple frequencies */
    Compute $\widehat{\boldsymbol{V}}^{\max,t+1} \in \mathbb{R}^{N \times M}$, whose $(i,m)$th entry satisfies

        $\widehat{\boldsymbol{V}}_{im}^{\max,t+1} \leftarrow \max_{\theta_i \in \Omega} \sum_{k=-K_{\max}}^{K_{\max}} \left\langle e^{\iota k \theta_i}, \widehat{\boldsymbol{V}}_{im}^{(k),t+1} \right\rangle \qquad (6.21)$

    /* Recovery of the cluster structure and associated phase angles
       */
    For each node $i \in [N]$, assign its cluster assignment as

        $\hat{\mathcal{M}}(i) \leftarrow \arg\max_{m \in [M]} \quad \widehat{\boldsymbol{H}}_{i,\cdot}^{t+1}, \text{ where } \widehat{\boldsymbol{H}}^{t+1} \leftarrow \mathcal{P}_{\mathcal{H}}(\widehat{\boldsymbol{V}}^{\max,t+1})$

      then estimate the associated phase angle given the estimated cluster
      assignment $\hat{\mathcal{M}}(i)$

        $\hat{\theta}_i \leftarrow \arg\max_{\theta_i \in \Omega} \sum_{k=-K_{\max}}^{K_{\max}} \left\langle e^{\iota k \theta_i}, \widehat{\boldsymbol{V}}_{i\hat{\mathcal{M}}(i)}^{(k),t+1} \right\rangle \qquad (6.22)$

    Construct $\{\widehat{\boldsymbol{V}}^{(k),t+1}\}_{k=-K_{\max}}^{K_{\max}}$ using $\{\hat{\mathcal{M}}(i)\}_{i=1}^N$ and $\{\hat{\theta}_i\}_{i=1}^N$ according
      to (6.10)
end
Output: Estimated cluster structure $\{\hat{\mathcal{M}}(i)\}_{i=1}^N$ and estimated phase angles
        $\{\hat{\theta}_i\}_{i=1}^N$
```

6.4 Iterative Multi-Frequency Generalized Power Method

In addition to the spectral method based on the MF-CPQR factorization proposed in Section 6.3, we develop an iterative multi-frequency generalized power method for the joint estimation problem which is inspired by the generalized power method [2] and the *"multi-frequency"* nature of the MLE formulation (6.9).

6.4.1 Detailed Steps and Motivations

Since the joint estimation problem is non-convex, the iterative multi-frequency generalized power method requires a good initialization of the cluster structure and associated phase angles that are sufficiently close to the ground truth. Various spectral algorithms (e.g., CPQR-type algorithm [1, Algorithm 1], [2, Algorithm 3], and Algorithm 6.1) can be used for initialization. It is observed experimentally that random initialization will result in convergence to a sub-optimal solution. Each iteration of Algorithm 6.4 consists of three main steps. The first step (line 3) is the matrix multiplication between $\boldsymbol{A}^{(k)}$ and $\widehat{\boldsymbol{V}}^{(k),t}$ for all $k = -K_{\max}, \ldots, K_{\max}$ (line 4). Then we leverage (line 4) $\widehat{\boldsymbol{V}}^{(k),t+1}$ across all frequencies to aggregate and refine the information needed for the joint estimation problem (6.21) which is inspired by (6.15). The last step is estimating the cluster structure and associated phase angles. As mentioned before, giving $\widehat{\boldsymbol{V}}^{\max,t+1}$ and then finding the corresponding cluster assignment is equal to solving the MCAP (see Definition 6.3). This is equivalent to projecting $\widehat{\boldsymbol{V}}^{\max,t+1}$ onto the feasible set $\mathcal{H}$ (line 5), after which the matrix $\widehat{\boldsymbol{H}}^{t+1}$ is obtained. The reason why the projection $\mathcal{P}(\cdot)$ is needed rather than directly using the index of the largest entry in each row of $\widehat{\boldsymbol{V}}^{\max,t+1}$ is because the solution of the latter approach does not necessarily satisfy the constraint based on the size of each cluster. The associated phase angles can be recovered according to the recovered cluster structure (6.22). Besides, the modification of the iterative MF-GPM for the joint estimation problem with continuous phase angles is the same as that of the spectral method based on the MF-CPQR factorization.

The iterative GPM in [118, Liu et al., 20] is built upon the classical power method which is used to compute the leading eigenvectors of a matrix. The method in [118, Liu et al., 20] adds an important step: projection onto the feasible set that is induced by the constraints on the cluster structure and phase angles. The iterative MF-GPM introduced here takes a step further by not only taking advantage of the efficiency of the power method and the projection, but also leveraging the information across multiple frequencies. In Section 6.5, numerical experiments show that the iterative MF-GPM largely outperforms GPM [2].

Table 6.3: The computational complexity of Algorithm 6.4 in each step.

Steps	Computational Complexity
1. Initialization	$\mathcal{O}(\lvert\mathcal{E}\rvert)$
2. Matrix multiplication	$\mathcal{O}(K_{\max}\lvert\mathcal{E}\rvert)$
3. Combine information	$\mathcal{O}(NK_{\max}\log K_{\max})$
4. Estimation	$\mathcal{O}(N\log N)$
Total complexity	$\mathcal{O}(K_{\max}\lvert\mathcal{E}\rvert + N(\log N + K_{\max}\log K_{\max}))$

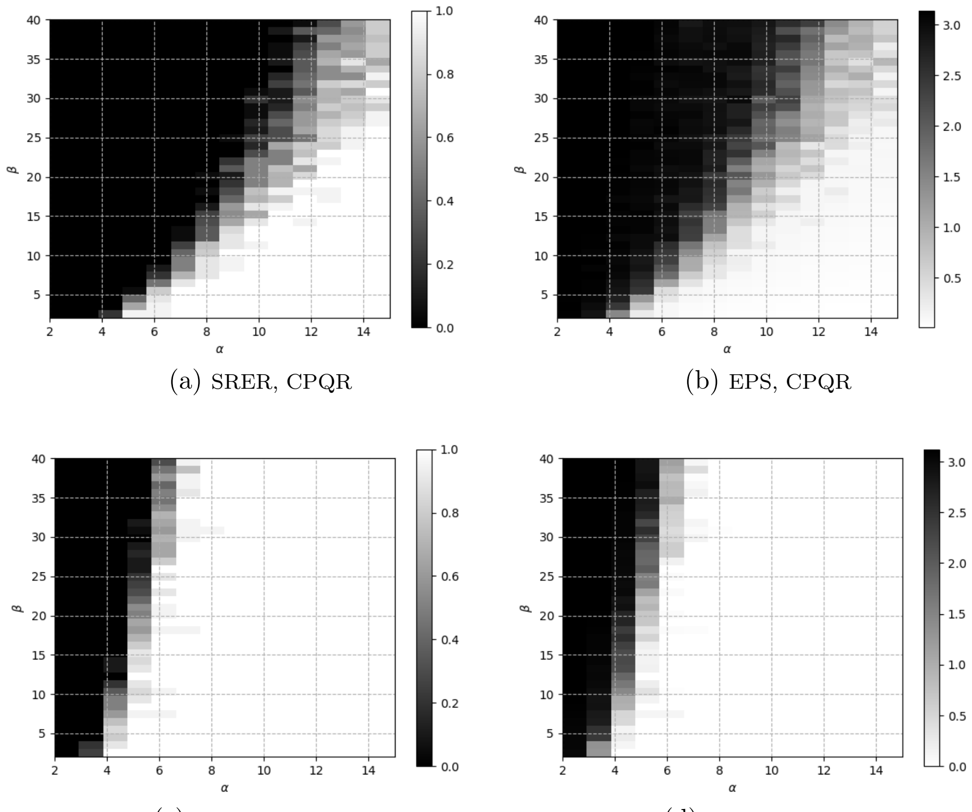
6.4.2 Computational Complexity

In this section, we compute the complexity of Algorithm 6.4 step by step in Table 6.3. Again, here we assume $M = \Theta(1)$. In terms of initialization, the CPQR-type algorithm [1] is $\mathcal{O}(|\mathcal{E}|)$. The matrix multiplication step consists of $\mathcal{O}(K_{\max})$ times of matrix multiplication ($\mathcal{O}(|\mathcal{E}|)$ per time). In order to combine information across multiple frequencies, we need to compute MN times of FFT of length-$K_{\max}$ vectors ($\mathcal{O}(K_{\max}\log K_{\max})$ per vector). For estimating cluster structure and associated phase angles, we first need to project $\widehat{\boldsymbol{V}}^{\max,t+1}$ onto $\mathcal{H}$, which is $\mathcal{O}(N\log N)$. Then complexity of estimating the cluster structure and associated phase angles using $\widehat{\boldsymbol{H}}^{t+1}$ is negligible. When the network $\mathcal{G}$ is densely connected with $|\mathcal{E}| = \mathcal{O}(N^2)$, Algorithm 6.4 ends up with $\mathcal{O}(K_{\max}N^2)$ if $N > \log K_{\max}$. However, if $|\mathcal{E}| = o(N^2)$, for example $\mathcal{O}(N\log N)$ and $\mathcal{O}(N)$, the complexity will be reduced to $\mathcal{O}(K_{\max}N\max\{\log N, \log K_{\max}\})$ and $\mathcal{O}(N\max\{\log N, K_{\max}\log K_{\max}\})$, respectively. As a result, the computational complexity of Algorithm 6.4 is very similar to Algorithm 6.1.

6.5 Numerical Experiments

This section deals with numerical experiments of the spectral method based on the MF-CPQR factorization (Algorithm 6.1) and the iterative MF-GPM (Algorithm 6.4) to showcase their performance against state-of-the-art benchmark algorithms[1]. For comparison, the benchmark algorithms are chosen as i) the CPQR-type algorithm [1], ii) the GPM [2], where both of them can be modified identically from the joint community and group synchronization problem into the joint community detection and phase synchronization

[1] Codes are available at `https://github.com/LingdaWang/Joint_Community_Detection_and_Phase_Synchronization`



(a) SRER, CPQR

(b) EPS, CPQR

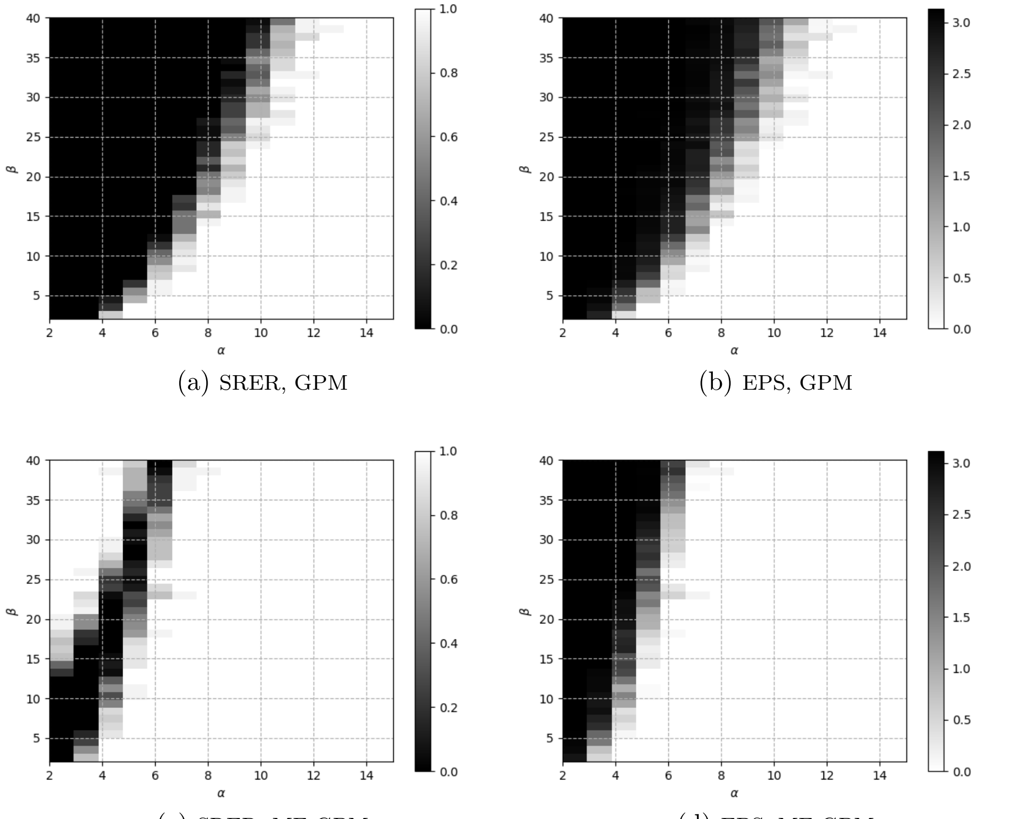
(c) SRER, MF-CPQR

(d) EPS, MF-CPQR

Figure 6.2: Comparison between the CPQR-type algorithm [1] (in the first row) and the spectral method based on the MF-CPQR factorization (in the second row) in terms of SRER and EPS, where a smaller black area in each figure indicates a better performance. Experiments are conducted with the setting $M = 2$, $N = 1000$, and $K_{\max} = 16$. (a) and (c): SRER (6.23) under varying α in $p = \alpha^{\log n}/n$ and β in $q = \beta^{\log n}/n$; (b) and (d): EPS (6.24) under varying α and β.

problem. Specifically, algorithms in [1, 2] are single frequency version of our proposed algorithms which can be realized by replacing the summation over k in (6.15), (6.16), (6.21), and (6.22) with $k = 1$.

In each experiment, we generate the observation matrix $\boldsymbol{A}$ using the probabilistic model, SBM-Ph, as discussed in Section 6.2 and estimate the cluster structure and associated phase angles by the spectral algorithms based on the MF-CPQR factorization, the iterative MF-GPM, and the benchmark algorithms. To evaluate the numerical results, we defined two metrics, *success rate of exact recovery* (SRER) and *error of phase synchronization* (EPS), for recovering the cluster structure and associated phase angles. In terms of



(a) SRER, GPM

(b) EPS, GPM

(c) SRER, MF-GPM

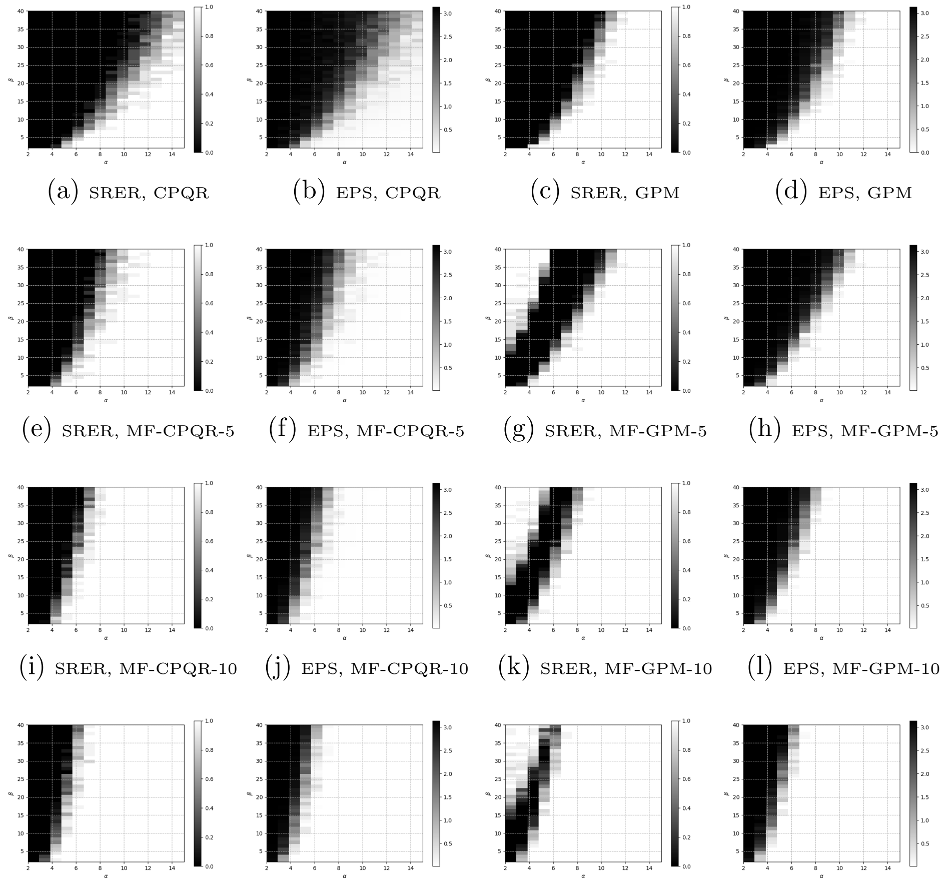
(d) EPS, MF-GPM

Figure 6.3: Comparison between the GPM [2] (in the first row) and the iterative MF-GPM (in the second row) in terms of SRER and EPS, where a smaller black area in each figure indicates a better performance. Experiments are conducted with the same setting as Figure 6.2. (a) and (c): SRER (6.23) under varying α in $p = \alpha^{\log n}/n$ and β in $q = \beta^{\log n}/n$; (b) and (d): EPS (6.24) under varying α and β.

SRER, it shows the rate of algorithms exactly recover the cluster structure. Let $\hat{\mathcal{S}}_m = \{i \in [N] | \hat{\mathcal{M}}(i) = m\}$ be the set of nodes assigned into the mth cluster by algorithms, and we have that

$$\text{SRER} = \text{ the rate } \{\hat{\mathcal{S}}_m\}_{m=1}^M \text{ is identical to } \{\mathcal{S}_m\}_{m=1}^M. \tag{6.23}$$

As for the EPS, it assesses the performance of recovering phase angles. We define $\boldsymbol{\theta}^{*,(m)} = [e^{\iota\theta_i^*}]_{i\in\mathcal{S}_m^*} \in \mathbb{C}^s$ for each cluster that concatenates the ground truth θ_i^* for all $i \in \mathcal{S}_m^*$, and similarly $\hat{\boldsymbol{\theta}}^{(m)} = [e^{\iota\hat{\theta}_i}]_{i\in\mathcal{S}_m^*} \in \mathbb{C}^s$ for the estimated phase angles. Then, after removing the ambiguity with aligning $\hat{\boldsymbol{\theta}}^{(m)}$ with



(a) SRER, CPQR (b) EPS, CPQR (c) SRER, GPM (d) EPS, GPM

(e) SRER, MF-CPQR-5 (f) EPS, MF-CPQR-5 (g) SRER, MF-GPM-5 (h) EPS, MF-GPM-5

(i) SRER, MF-CPQR-10 (j) EPS, MF-CPQR-10 (k) SRER, MF-GPM-10 (l) EPS, MF-GPM-10

(m) SRER, MF-CPQR-20 (n) EPS, MF-CPQR-20 (o) SRER, MF-GPM-20 (p) EPS, MF-GPM-20

Figure 6.4: Results for the joint estimation problem with continuous phase angles in $[0, 2\pi)$ using the CPQR-type algorithm [1], the GPM [2], the spectral method based on the MF-CPQR factorization, and the iterative MF-GPM, where a smaller black area in each figure indicates better performance. The choice of M and N are the same as Figure 6.2. The first and third columns show the SRER, and the second and fourth columns shows EPS. (a), (b), (c), and (d): The results of the CPQR-type algorithm [1] and the GPM [2]; (e), (f), (g), and (h): The results of the spectral method based on the MF-CPQR factorization and the iterative MF-GPM with $K_{\max} = 5$; (i), (j), (k), and (l): The results of the spectral method based on the MF-CPQR factorization and the iterative MF-GPM with $K_{\max} = 10$; (m), (n), (o), and (p): The results of the spectral method based on the MF-CPQR factorization and the iterative MF-GPM with $K_{\max} = 20$.

$\boldsymbol{\theta}^{*,(m)}$ in each cluster as

$$\gamma^{(m)} = \underset{g^{(m)} \in \Omega \text{ or } [0,2\pi)}{\arg\min} \quad \|\hat{\boldsymbol{\theta}}^{(m)} e^{\iota g^{(m)}} - \boldsymbol{\theta}^{*,(m)}\|_2, \quad \forall m = 1, \ldots, M,$$

the EPS is defined as

$$\text{EPS} = \max_{m\in[M]} \max_{i\in\mathcal{S}_m^*} \{\min(|\hat{\theta}_i + \gamma^{(m)} - \theta_i^*|, 2\pi - |\hat{\theta}_i + \gamma^{(m)} - \theta_i^*|)\}. \tag{6.24}$$

The EPS is actually the maximum error of estimated phase angles among all nodes. Besides, both the SRER and EPS are computed over 20 independent and identical realizations for each experiment in the following. In the rest of this section, we first present the results of the joint estimation problem in Section 6.5.1 and followed by the extension to continuous phase angles in Section 6.5.2.

6.5.1 Results of the Joint Estimation Problem

We first show the results of the spectral method based on the MF-CPQR factorization (Algorithm 6.1) against the CPQR-type algorithm [1] on the joint estimation problem, where the case of $M = 2$, $s = 500$, and $K_{\max} = 16$ is considered. Similar to [1, 2], we test the recovery performance in the regime $p, q = \mathcal{O}({}^{\log n}/_{n})$, where different $p = \alpha\,{}^{\log n}/_{n}$ and $q = \beta\,{}^{\log n}/_{n}$ with varying α and β are included. In Figure 6.2, we show SRER (6.23) and EPS (6.24). As one can observe from Figure 6.2a and 6.2c, our proposed spectral method based on the MF-CPQR factorization outperforms the CPQR-type algorithm [1] in SRER. EPS follows a similar pattern.

Next, we test the performance of the iterative MF-GPM (Algorithm 6.4) against the GPM [2] under the same choice of M, s, and $K_{\max}$ as before. Since the GPM and the iterative MF-GPM require initialization that is close enough to the ground truth, we can choose either [2, Algorithm 3] or the CPQR-type algorithm [1]. We set the number of iterations to be 50 as suggested by [2]. Again, as one can observe from Figure 6.3, our proposed iterative MF-GPM achieves higher accuracy in both SRER and EPS. Surprisingly, one may also notice the region where p is small and q is large (top left area in Figure 6.3c), the iterative MF-GPM is capable of recovering the cluster structure with high probability: however, this is not the case in recovering associated phase angles.

When comparing the results shown in Figure 6.2 and 6.3 together, the spectral method based on the MF-CPQR factorization shows very similar result as the iterative MF-GPM which are both significantly better than the

GPM [2] and the CPQR-type algorithm [1]. However, compared to the iterative MF-GPM, the spectral method based on the MF-CPQR factorization is free of initialization. One may also observe the performance of the GPM [2] outperform the CPQR-type algorithm [1].

6.5.2 Results with Continuous Phase Angles

In this section, we show the results of our proposed algorithms against benchmark algorithms on the joint estimation problem with continuous phase angles. As mentioned in Section 6.2.3, the algorithms tested in Section 6.5.1 can be directly applied after simple modification (See Section 6.3.1 for details), and thus we choose the similar setting as Section 6.5.1. Besides, since (6.13) is a truncated MLE formulation of the true one (6.11), experiments of the spectral method based on the MF-CPQR factorization and the iterative MF-GPM with different $K_{\max}$ are conducted to study the trend of the results as $K_{\max}$ grows. The results are detailed in Figure 6.4 with very similar performance as shown in Figure 6.2 and 6.3. In addition, as $K_{\max}$ grows, the cluster structure recovery and phase synchronization become more accurate in both MF-CPQR based method and iterative MF-GPM.

To choose a suitable $K_{\max}$ for the continuous phase angles, we need to consider the trade-off between the performance and the computational complexity. We observe that the estimation accuracy is improved as $K_{\max}$ increases. On the other hand, the computational complexity scales linearly with $K_{\max}$. In addition, the computational complexity also depends on the number of nodes N and the number of clusters M, which needs to be taken into consideration for the trade-off between accuracy and efficiency. Thus, it is difficult to state a simple optimal policy for choosing $K_{\max}$ for the continuous phase angles. Despite this, we have shown that our methods outperform the CPQR-type algorithm and the GPM as long as $K_{\max} \geq 1$, and moreover largely outperform other baseline algorithms when $K_{\max} \geq 10$. Therefore, our choice of $K_{\max}$ is between 10 to 30 for most cases.

Part III

Accelerated Methods for Convex Optimization Problems

CHAPTER 7

ALMOST TUNE FREE VARIANCE REDUCTION

7.1 Preliminaries

Notation. In Chapter 7, bold lowercase letters denote column vectors; $\mathbb{E}$ represents expectation; $\|\mathbf{x}\|$ stands for the ℓ_2-norm of $\mathbf{x}$; and $\langle \mathbf{x}, \mathbf{y} \rangle$ denotes the inner product of vectors $\mathbf{x}$ and $\mathbf{y}$.

We will first focus on the averaging techniques, whose generality goes beyond BB step sizes. To start with, this section briefly reviews the vanilla SVRG and SARAH while their BB variants are postponed slightly.

7.1.1 Basic Assumptions

Assumption 7.1. *Each $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ has L-Lipchitz gradient, that is, $\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.*

Assumption 7.2. *Each $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ is convex.*

Assumption 7.3. *Function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is μ-strongly convex, that is, there exists $\mu > 0$, such that $f(\mathbf{x}) - f(\mathbf{y}) \geq \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2$, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.*

Assumption 7.4. *Each $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ is μ-strongly convex, meaning there exists $\mu > 0$, so that $f_i(\mathbf{x}) - f_i(\mathbf{y}) \geq \langle \nabla f_i(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2$, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.*

Assumption 7.1 requires each loss function to be sufficiently smooth. One can certainly require smoothness of each individual loss function and refine Assumption 7.1 as f_i has L_i-Lipchitz gradient. Clearly $L = \max_i L_i$. By combining with importance sampling [235, 236, 237], such a refined assumption can slightly tighten the κ dependence in the complexity bound. However, since the extension is straightforward, we will keep using the simpler Assumption 7.1 for clarity. Assumption 7.3 only requires f to be strongly convex which is weaker than Assumption 7.4. Assumptions 7.1 – 7.4 are all standard in variance reduction algorithms.

7.1.2 Recap of SVRG and SARAH

Algorithm 7.1: SVRG

```
Initialize: $\tilde{\mathbf{x}}^0$, $\eta$, $m$, $S$
for $s = 1, 2, \ldots, S$ do
    $\mathbf{x}_0^s = \tilde{\mathbf{x}}^{s-1}$
    $\mathbf{g}^s = \nabla f(\mathbf{x}_0^s)$
    for $k = 0, 1, \ldots, m-1$ do
        uniformly draw $i_k \in [n]$
        $\mathbf{v}_k^s = \nabla f_{i_k}(\mathbf{x}_k^s) - \nabla f_{i_k}(\mathbf{x}_0^s) + \mathbf{g}^s$
        $\mathbf{x}_{k+1}^s = \mathbf{x}_k^s - \eta \mathbf{v}_k^s$
    end for
    select $\tilde{\mathbf{x}}^s$ randomly from $\{\mathbf{x}_k^s\}_{k=0}^m$ following $\mathbf{p}^s$
end for
Output: $\tilde{\mathbf{x}}^S$
```

Algorithm 7.2: SARAH

```
Initialize: $\tilde{\mathbf{x}}^0$, $\eta$, $m$, $S$
for $s = 1, 2, \ldots, S$ do
    $\mathbf{x}_0^s = \tilde{\mathbf{x}}^{s-1}$, and $\mathbf{v}_0^s = \nabla f(\mathbf{x}_0^s)$
    $\mathbf{x}_1^s = \mathbf{x}_0^s - \eta \mathbf{v}_0^s$
    for $k = 1, 2, \ldots, m-1$ do
        uniformly draw $i_k \in [n]$
        $\mathbf{v}_k^s = \nabla f_{i_k}(\mathbf{x}_k^s) - \nabla f_{i_k}(\mathbf{x}_{k-1}^s) + \mathbf{v}_{k-1}^s$
        $\mathbf{x}_{k+1}^s = \mathbf{x}_k^s - \eta \mathbf{v}_k^s$
    end for
    select $\tilde{\mathbf{x}}^s$ randomly from $\{\mathbf{x}_k^s\}_{k=0}^m$ following $\mathbf{p}^s$
end for
Output: $\tilde{\mathbf{x}}^S$
```

The steps of SVRG and SARAH are listed in Algorithm 7.1 and 7.2, respectively. Each employs a fine-grained reduced-variance gradient estimate per iteration. For SVRG, $\mathbf{v}_k^s$ is an unbiased estimate since $\mathbb{E}[\mathbf{v}_k^s|\mathcal{F}_{k-1}^s] = \nabla f(\mathbf{x}_k^s)$, where $\mathcal{F}_{k-1}^s := \sigma(\tilde{\mathbf{x}}^{s-1}, i_0, i_1, \ldots, i_{k-1})$ is the σ-algebra generated by $\tilde{\mathbf{x}}^{s-1}, i_1, i_2, \ldots, i_{k-1}$; while SARAH adopts a biased $\mathbf{v}_k^s$, that is, $\mathbb{E}[\mathbf{v}_k^s|\mathcal{F}_{k-1}^s] =$

$\nabla f(\mathbf{x}_k^s) - \nabla f(\mathbf{x}_{k-1}^s) + \mathbf{v}_{k-1}^s \neq \nabla f(\mathbf{x}_k^s)$. The variance (mean-square error (MSE)) of $\mathbf{v}_k^s$ in SVRG (SARAH) can be upper bounded by quantities that dictate the optimality gap (gradient norm square).

Lemma 7.1. *[128, 132] The MSE of $\mathbf{v}_k^s$ in SVRG is bounded as follows:*

$$\begin{aligned}\text{SVRG}: \mathbb{E}\big[\|\nabla f(\mathbf{x}_k^s) - \mathbf{v}_k^s\|^2\big] &\leq \mathbb{E}\big[\|\mathbf{v}_k^s\|^2\big] \\ &\leq 4L\mathbb{E}\big[f(\mathbf{x}_k^s) - f(\mathbf{x}^*)\big] + 4L\mathbb{E}\big[f(\mathbf{x}_0^s) - f(\mathbf{x}^*)\big].\end{aligned}$$

The MSE of $\mathbf{v}_k^s$ in SARAH is also bounded as

$$\text{SARAH}: \quad \mathbb{E}\big[\|\nabla f(\mathbf{x}_k^s) - \mathbf{v}_k^s\|^2\big] \leq \frac{\eta L}{2 - \eta L}\bigg(\mathbb{E}\big[\|\nabla f(\mathbf{x}_0^s)\|^2\big] - \mathbb{E}\big[\|\mathbf{v}_k^s\|^2\big]\bigg).$$

Another upper bound on SVRG's gradient estimate is available; see e.g., [236], but it is not suitable for our analysis. Intuitively, Lemma 7.1 suggests that if SVRG or SARAH converges, the MSE of their gradient estimates also approaches to zero.

At the end of each inner loop, the starting point of the next outer loop is randomly selected among $\{\mathbf{x}_k^s\}_{k=0}^m$ according to a pmf vector $\mathbf{p}^s \in \Delta_{m+1}$, where $\Delta_{m+1} := \{\mathbf{p} \in \mathbb{R}_+^{m+1} | \langle \mathbf{1}, \mathbf{p}\rangle = 1\}$. We term $\mathbf{p}^s$ the *averaging weight vector*, and let p_j^s denote the jth entry of $\mathbf{p}^s$. Leveraging the MSE bounds in Lemma 7.1 and choosing a proper averaging vector, SVRG and SARAH iterates for strongly convex problems can be proved to converge linearly.

For SVRG, two types of averaging exist.

- **U-Avg (SVRG)** [128]: vector $\mathbf{p}^s$ is chosen as the pmf of an (almost) uniform distribution; that is, $p_m^s = 0$, and $p_k^s = 1/m$ for $k = \{0, 1, \ldots, m-1\}$. Under Assumptions 7.1 – 7.3, the choice of $\eta = \mathcal{O}(1/L)$ and $m = \mathcal{O}(\kappa)$ ensures that SVRG iterates converge linearly.[1]

- **L-Avg (SVRG)** [140, 238]: Only the last iteration is used for averaging by setting $\tilde{\mathbf{x}}^s = \mathbf{x}_m^s$; or equivalently, by setting $p_m^s = 1$, and $p_k^s = 0, \forall k \neq m$. Under Assumptions 7.1 – 7.3, linear convergence is ensured by choosing $\eta = \mathcal{O}(1/(L\kappa))$ and $m = \mathcal{O}(\kappa^2)$.

[1] For simplicity and clarity of exposition we only highlight the order of η and m, and hide other constants in big-$\mathcal{O}$ notation. Detailed choices can be found in the corresponding references.

To guarantee linear convergence, SVRG with L-Avg must adopt a much smaller η and larger m compared with U-Avg. L-Avg with such a small step size leads to complexity $\mathcal{O}\big((n+\kappa^2)\ln\frac{1}{\epsilon}\big)$ that has worse dependence on κ.

For SARAH, there are also two averaging options.

- **U-Avg (SARAH)** [132]: here $\mathbf{p^s}$ is selected to have entries $p_m^s = 0$, and $p_k^s = 1/m$, for $k = \{0, 1, \ldots, m-1\}$. Linear convergence is guaranteed with complexity $\mathcal{O}\big((n+\kappa)\ln\frac{1}{\epsilon}\big)$ under Assumptions 7.1 – 7.3 so long as one selects $\eta = \mathcal{O}(1/L)$ and $m = \mathcal{O}(\kappa)$.

- **L-Avg (SARAH)** [136]: here $\mathbf{p}^s$ is chosen with entries $p_{m-1}^s = 1$ and $p_k^s = 0, \forall k \neq m-1$. Under Assumptions 7.1 – 7.3 and with $\eta = \mathcal{O}(1/L)$ as well as $m = \mathcal{O}(\kappa^2)$, linear convergence is guaranteed at complexity of $\mathcal{O}\big((n+\kappa^2)\ln\frac{1}{\epsilon}\big)$. When both Assumptions 7.1 and 7.4 hold, setting $\eta = \mathcal{O}(1/L)$ and $m = \mathcal{O}(\kappa)$ results in linear convergence along with a reduced complexity of order $\mathcal{O}\big((n+\kappa)\ln\frac{1}{\epsilon}\big)$.

U-Avg (for both SVRG and SARAH) is usually employed as a "proof-trick" to carry out convergence analysis, while L-Avg is implemented most of the times. However, we will argue in the next section that with U-Avg adapted to the step size choice, it is possible to improve empirical performance. Although U-Avg appears at first glance to waste updates, a simple trick in the implementation can fix this issue.

Implementation of Averaging. Rather than updating m times and then choosing $\tilde{\mathbf{x}}^s$ according to Line 10 of SVRG or SARAH, one can generate a random integer $M^s \in \{0, 1, \ldots, m\}$ according to the averaging weight vector $\mathbf{p}^s$. Having available $\mathbf{x}_{M^s}^s$, it is possible to start the next outer loop immediately.

7.2 Weighted Averaging for SVRG and SARAH

This section introduces weighted averaging for SVRG and SARAH which serves as an intermediate step for the ultimate tune-free variance reduction. Such an averaging for SVRG will considerably tighten its analytical convergence rate, while for SARAH it will improve its convergence rate when m or η is chosen sufficiently large. These analytical results are obtained by reexamining SVRG and SARAH through the estimate sequence (ES), a tool that has been used for analyzing momentum schemes [125]; see also [239, 240, 236]. Different from existing ES analysis that relies heavily on the unbiasedness of $\mathbf{v}_k^s$, our advances here will endow ES with the ability to deal with the biased gradient estimate of SARAH.

7.2.1 Estimate Sequence

Since in this section we will focus on a specific inner loop indexed by s, the superscript s is dropped for brevity. For example, $\mathbf{x}_k^s$ and $\mathbf{v}_k^s$ are written as $\mathbf{x}_k$ and $\mathbf{v}_k$, respectively.

Associated with the ERM objective f and a particular point $\mathbf{x}_0$, consider a series of quadratic functions $\{\Phi_k(\mathbf{x})\}_{k=0}^m$ that comprise what is termed ES, with the first one given by

$$\Phi_0(\mathbf{x}) = \Phi_0^* + \frac{\mu_0}{2}\|\mathbf{x} - \mathbf{x}_0\|^2, \tag{7.2a}$$

and the rest defined recursively as

$$\Phi_k(\mathbf{x}) =(1-\delta_k)\Phi_{k-1}(\mathbf{x}) + \delta_k\Big[f(\mathbf{x}_{k-1}) + \langle \mathbf{v}_{k-1}, \mathbf{x} - \mathbf{x}_{k-1}\rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{x}_{k-1}\|^2\Big],$$

where $\mathbf{v}_{k-1}$ is the gradient estimate in SVRG or SARAH while Φ_0^*, μ_0, and δ_k are some constants to be specified later. The design is similar to that of [236], but the ES here is constructed per inner loop. In addition, here we will overcome the challenge of analyzing SARAH's biased gradient estimate $\mathbf{v}_k$.

Upon defining $\Phi_k^* := \min_{\mathbf{x}} \Phi_k(\mathbf{x})$, the key properties of the sequence $\{\Phi_k(\mathbf{x})\}_{k=0}^m$ are collected in the next lemma.

Lemma 7.2. *For $\{\Phi_k(\mathbf{x})\}_{k=0}^m$ as in (7.2), it holds that: i) $\Phi_0(\mathbf{x})$ is μ_0-strongly convex, and $\Phi_k(\mathbf{x})$ is μ_k-strongly convex with $\mu_k = (1-\delta_k)\mu_{k-1}+\delta_k\mu$; ii) $\mathbf{x}_k$ minimizes $\Phi_k(\mathbf{x})$ if $\delta_k = \eta\mu_k$; and iii) $\Phi_k^* = (1-\delta_k)\Phi_{k-1}^* + \delta_k f(\mathbf{x}_{k-1}) - \frac{\mu_k\eta^2}{2}\|\mathbf{v}_{k-1}\|^2$.*

Lemma 7.2 holds for both SVRG and SARAH. To better understand the role of ES, it is instructive to use an example.

Example. With $\Phi_0^* = f(\mathbf{x}_0)$, $\mu_0 = \mu$, and $\delta_k = \mu_k\eta$ for SVRG, it holds that $\mu_k = \mu, \forall k$, and $\delta_k = \mu\eta, \forall k$. If for convenience we let $\delta := \mu\eta$, we show in

Section 7.5.2 that

$$\mathbb{E}\big[\Phi_k(\mathbf{x})\big] \leq (1-\delta)^k\big[\Phi_0(\mathbf{x}) - f(\mathbf{x}^*)\big] + f(\mathbf{x}). \tag{7.3}$$

As $k \to \infty$, one has $(1-\delta)^k \to 0$, and hence $\Phi_k(\mathbf{x})$ approaches in expectation a lower bound of $f(\mathbf{x})$.

Now, we are ready to view SVRG and SARAH through the lens of $\{\Phi_k(\mathbf{x})\}_{k=0}^m$ to obtain new averaging schemes.

7.2.2 Weighted Averaging for SVRG

The new averaging vector $\mathbf{p}^s$ for SVRG together with the improved convergence rate is summarized in the following theorem.

Theorem 7.1. ***(SVRG with W-Avg.)*** *Under Assumptions 7.1 – 7.3, construct the ES as in* (7.2) *with* $\mu_0 = \mu$, $\delta_k = \mu_k \eta$, *and* $\Phi_0^* = f(\mathbf{x}_0)$. *Choose* $\eta < 1/(4L)$, *and* m *large enough such that*

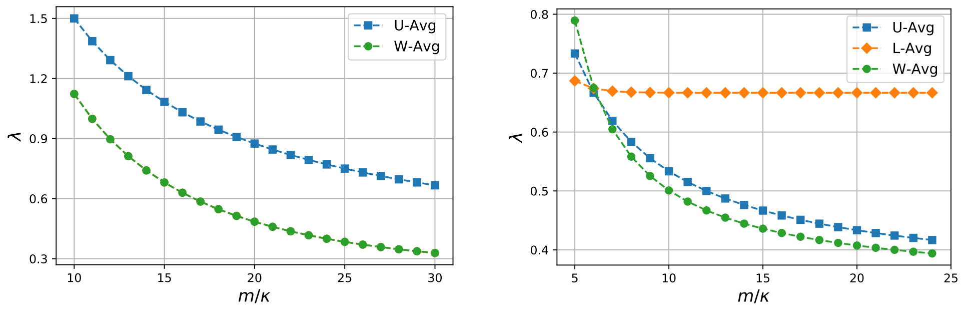
$$\lambda^{\textit{SVRG}} := \frac{1}{1-(1-\mu\eta)^{m-1}}\left[\frac{(1-\mu\eta)^m}{1-2\eta L} + \frac{2\mu L\eta^2(1-\mu\eta)^{m-1}}{1-2L\eta} + \frac{2L\eta}{1-2L\eta}\right] < 1.$$

Let $p_0^s = p_m^s = 0$, *and* $p_k^s = (1-\mu\eta)^{m-k-1}/q$ *for* $k = 1, 2, \ldots, m-1$, *where* $q = [1-(1-\mu\eta)^{m-1}]/(\mu\eta)$. *It then holds for SVRG with this weighted averaging (W-Avg) that*

$$\mathbb{E}\big[f(\tilde{\mathbf{x}}^s) - f(\mathbf{x}^*)\big] \leq \lambda^{\textit{SVRG}}\mathbb{E}\big[f(\tilde{\mathbf{x}}^{s-1}) - f(\mathbf{x}^*)\big].$$

Comparing the W-Avg in Theorem 7.1 against U-Avg and L-Avg, we saw in Section 7.1.2, the upshot of W-Avg is a much tighter convergence rate. When choosing $\eta = \mathcal{O}(1/L)$, the dominating terms of the convergence rate for W-Avg are $\mathcal{O}\big(\frac{(1-1/\kappa)^m}{1-2L\eta} + \frac{2L\eta}{1-2L\eta}\big)$, and $\mathcal{O}\big(\frac{\kappa}{m(1-2L\eta)} + \frac{2L\eta}{1-2L\eta}\big)$ for U-Avg [128]. Clearly, the factor $(1-1/\kappa)^m$ in W-Avg can be much smaller than κ/m in U-Avg; see Figure 7.1(a) for comparison of convergence rates of different averaging types. Since convergence of SVRG with L-Avg requires η and m to be chosen differently from those in U-Avg and W-Avg, L-Avg is not plotted in Figure 7.1(a).

Next, we assess the complexity of SVRG with W-Avg.



(a) SVRG

(b) SARAH

Figure 7.1: A comparison of the analytical convergence rate for SVRG and SARAH. In both figures we set $\kappa = 10^5$ with $L = 1$, $\mu = 10^{-5}$, and the step sizes are selected as: (a) SVRG with $\eta = 0.1/L$; and (b) SARAH with $\eta = 0.5/L$.

Corollary 7.1. *Choosing $m = \mathcal{O}(\kappa)$ and other parameters as in Theorem 7.1, the complexity of SVRG with W-Avg to find $\tilde{\mathbf{x}}^s$ satisfying $\mathbb{E}\big[f(\tilde{\mathbf{x}}^s) - f(\mathbf{x}^*)\big] \leq \epsilon$ is $\mathcal{O}\big((n+\kappa)\ln\frac{1}{\epsilon}\big)$.*

Note that similar to U-Avg, W-Avg incurs lower complexity compared with L-Avg.

7.2.3 Weighted Averaging for SARAH

SARAH is challenging to analyze due to the bias present in the estimate $\mathbf{v}_k$, which makes the ES-based treatment of SARAH fundamentally different from that of SVRG. To see this, it is useful to start with the following lemma.

Lemma 7.3. *For any deterministic $\mathbf{x}$, it holds in SARAH that*

$$\begin{aligned}
&\mathbb{E}\big[\langle \mathbf{v}_k - \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k\rangle\big] \\
&\qquad = \frac{\eta}{2}\sum_{\tau=0}^{k-1}\mathbb{E}\Big[\|\mathbf{v}_\tau - \nabla f(\mathbf{x}_\tau)\|^2 + \|\mathbf{v}_\tau\|^2 - \|\nabla f(\mathbf{x}_\tau)\|^2\Big].
\end{aligned}$$

Lemma 7.3 reveals the main difference in the ES-based argument for SARAH, namely that $\mathbb{E}\big[\langle \mathbf{v}_k - \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k\rangle\big] \neq 0$, while the same inner product for SVRG equals to 0 in expectation. Reflecting back to (7.3), the consequence of having a non-zero $\mathbb{E}\big[\langle \mathbf{v}_k - \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k\rangle\big]$ is that $\mathbb{E}[\Phi_k(\mathbf{x})]$

is not necessarily approaching a lower bound of $f(\mathbf{x})$ as $k \to \infty$; thus,

$$\mathbb{E}\big[\Phi_k(\mathbf{x})\big] \leq (1-\delta)^k\big[\Phi_0(\mathbf{x}) - f(\mathbf{x})\big] + f(\mathbf{x}) + C, \tag{7.4}$$

where C is a non-zero term that is not present in (7.3) when applied to SVRG; see detailed derivations in Section 7.5.2.

Interestingly, upon capitalizing on the properties of $\mathbf{v}_k$, the ensuing theorem establishes linear convergence for SARAH with a proper W-Avg vector $\mathbf{p}^s$.

Theorem 7.2. ***(SARAH with W-Avg.)*** *Under Assumptions 7.1 and 7.4, define the ES as in* (7.2) *with $\mu_0 = \mu$, $\delta_k = \mu_k\eta, \forall k$, and $\Phi_0^* = f(\mathbf{x}_0)$. With $\delta := \mu\eta$, select $\eta < 1/L$ and m large enough, so that*

$$\begin{aligned}\lambda^{\textit{SARAH}} := &\left[(1-\delta)^m - \Big(1-\frac{2\eta L}{1+\kappa}\Big)^m\right]\frac{L+\mu}{c(L-\mu)}\\ &+\frac{(1-\delta)^m}{c\delta} + \frac{\eta L(m-1)}{c(2-\eta L)} + \frac{2-2\eta L}{2-\eta L}\frac{1+\kappa}{2c\eta L} < 1,\end{aligned}$$

where $c = m - \frac{1}{\delta} + \frac{(1-\delta)^m}{\delta}$. Setting $p_k = (1-(1-\delta)^{m-k-1})/c, \forall k = 0, 1, \ldots, m-2$, and $p_{m-1} = p_m = 0$, SARAH with this W-Avg satisfy

$$\mathbb{E}\big[\|\nabla f(\tilde{\mathbf{x}}^s)\|^2\big] \leq \lambda^{\textit{SARAH}}\mathbb{E}\big[\|\nabla f(\tilde{\mathbf{x}}^{s-1})\|^2\big].$$

The expression of λ^{SARAH} is complicated because we want the upper bound of the convergence rate to be as tight as possible. To demonstrate this with an example, choosing $\eta = 1/(2L)$ and $m = 5\kappa$, we have $\lambda^{\mathsf{SARAH}} \approx 0.8$. Figure 7.1(b) compares SARAH with W-Avg versus SARAH with U-Avg and L-Avg. The advantage of W-Avg is more pronounced as m is chosen larger.

As far as complexity of SARAH with W-Avg, it is comparable with that of L-Avg or U-Avg, as asserted next.

Corollary 7.2. *Choosing $m = \mathcal{O}(\kappa)$ and other parameters as in Theorem 7.2, the complexity of SARAH with W-Avg to find $\tilde{\mathbf{x}}^s$ satisfying $\mathbb{E}[\|\nabla f(\tilde{\mathbf{x}}^s)\|^2] \leq \epsilon$, is $\mathcal{O}\big((n+\kappa)\ln\frac{1}{\epsilon}\big)$.*

A few remarks are now in order on our analytical findings: i) most existing ES-based proofs use $\mathbb{E}[f(\tilde{\mathbf{x}}^s) - f(\mathbf{x}^*)]$ as optimality metric, while Theorem 7.2 and Corollary 7.2 rely on $\mathbb{E}[\|\nabla f(\tilde{\mathbf{x}}^s)\|^2]$; ii) the analysis method still holds

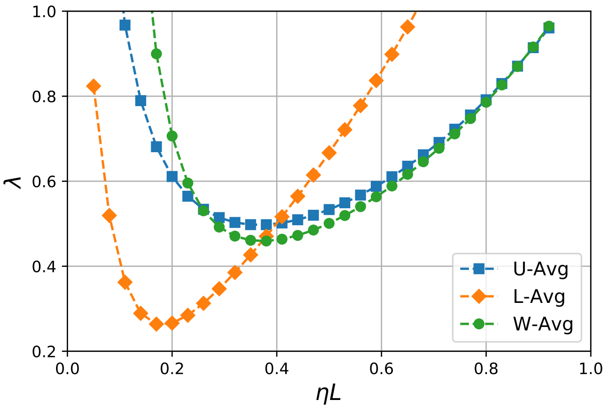


Figure 7.2: SARAH's analytical convergence with different averaging options ($\kappa = 10^5$, $L = 1$, $\mu = 10^{-5}$, and fixed $m = 10\kappa$).

when Assumption 7.4 is weakened to Assumption 7.3, at the price of having worse κ-dependence of the complexity, that is, $\mathcal{O}\big((n + \kappa^2) \ln \frac{1}{\epsilon}\big)$, which is of the same order as L-Avg under Assumptions 7.1 – 7.3 [136].

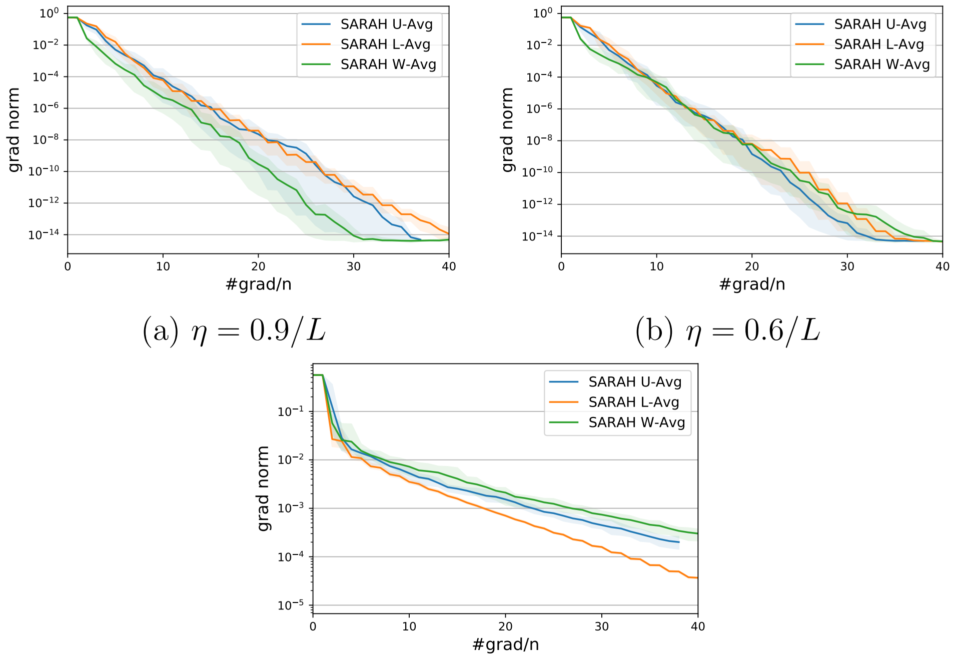
7.2.4 Averaging Is More Than A "Proof Trick"

Existing forms of averaging such as U-Avg and W-Avg are typically considered "proof tricks" for simplifying the theoretical analysis [128, 132, 136]. In this section, we contend that averaging can distinctly affect performance and should be adapted to other parameters. We will take SARAH with $\eta = \mathcal{O}(1/L)$ and $m = \mathcal{O}(\kappa)$ as an example, rather than SVRG since such parameter choices guarantee convergence regardless of the averaging employed. For SVRG with L-Avg on the other hand, the step size has to be chosen differently with W-Avg or U-Avg.

We will first look at the convergence rate of SARAH across different averaging options. Fixing $m = \mathcal{O}(\kappa)$ and changing η, the theoretical convergence rate is plotted in Figure 7.2. It is observed that with smaller step sizes, L-Avg enjoys faster convergence, while larger step sizes tend to favor W-Avg and U-Avg instead.

Next, we will demonstrate empirically that the type of averaging indeed matters. Consider binary classification using the regularized logistic loss function

$$f(\mathbf{x}) = \frac{1}{n} \sum_{i \in [n]} \ln \left[1 + \exp(-b_i \langle \mathbf{a}_i, \mathbf{x} \rangle)\right] + \frac{\mu}{2} \|\mathbf{x}\|^2, \tag{7.5}$$



(a) $\eta = 0.9/L$

(b) $\eta = 0.6/L$

(c) $\eta = 0.06/L$

Figure 7.3: Comparing SARAH with different types of averaging on dataset *w7a* ($\mu = 0.005$ and $m = 5\kappa$ in all experiments).

where $(\mathbf{a}_i, b_i)$ is the (feature, label) pair of datum i. Clearly, (7.5) is an instance of the cost in (1.3) with $f_i(\mathbf{x}) = \ln\big[1 + \exp(-b_i\langle\mathbf{a}_i, \mathbf{x}\rangle)\big] + \frac{\mu}{2}\|\mathbf{x}\|^2$; it can be readily verified that Assumptions 7.1 and 7.4 are satisfied in this case.

SARAH with L-Avg, U-Avg, and W-Avg are tested with fixed (moderate) $m = \mathcal{O}(\kappa)$ but different step size choices on the dataset *w7a*; see also Section 7.8.1 for additional experiments with datasets *a9a* and *diabetes*. Figure 7.3(a) shows that for a large step size $\eta = 0.9/L$, W-Avg outperforms U-Avg as well as L-Avg by almost two orders at the 30th sample pass. For a medium step size $\eta = 0.6/L$, W-Avg and L-Avg perform comparably, while both are outperformed by U-Avg. When η is chosen small, L-Avg is clearly the winner. In short, the performance of averaging options varies with the step sizes. This is intuitively reasonable because the MSE of $\mathbf{v}_k$: i) scales with η (cf. Lemma 7.1), and ii) tends to increase with k as $\mathbb{E}[\|\mathbf{v}_k\|^2]$ decreases linearly (see Lemma 7.5 in Section 7.6.2 and the MSE bound in Lemma 7.1). As a result, when both η and k are large, the MSE of $\mathbf{v}_k$ tends to be large too. Iterates with gradient estimates having high MSE can jeopardize the convergence.

This explains the inferior performance of L-Avg in Figure 7.3(a) and 7.3(b). On the other hand, when η is chosen small, the MSE tends to be small as well; hence, working with L-Avg does not compromise convergence, while in expectation W-Avg and U-Avg compute full gradient more frequently than L-Avg. These two reasons explain the improved performance of L-Avg in Figure 7.3(c).

When we fix η and change m, as depicted in Figure 7.1(b), the analytical convergence rate of W-Avg improves over that of U-Avg and L-Avg when m is large. This is because the MSE of $\mathbf{v}_k$ increases with k. W-Avg and U-Avg ensure better performance through "early ending", by reducing the number of updates that utilize $\mathbf{v}_k$ with large MSE.

In sum, the choice of averaging scheme should be adapted with η and m to optimize performance. For example, the proposed W-Avg for SARAH favors the regime where either η or m is chosen large, as dictated by the convergence rates and corroborated by numerical experiments.

7.3 Tune-Free Variance Reduction

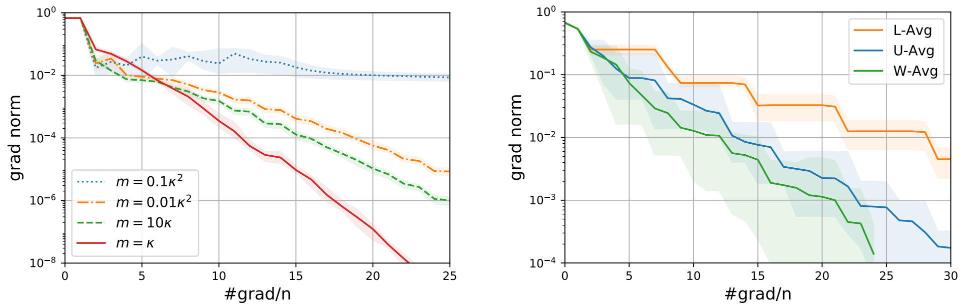
This section copes with variance reduction without tuning. In particular, i) BB step size, ii) averaging schemes, and iii) a time varying inner loop length are adopted for the best empirical performance.

7.3.1 Recap of BB Step Sizes

Aiming to develop "tune-free" SVRG and SARAH, we will first adopt the BB scheme to obtain suitable step sizes automatically. In a nutshell, BB monitors progress of previous outer loops and chooses the step size of outer loop s accordingly via

$$\eta^s = \frac{1}{\theta_\kappa} \frac{\|\tilde{\mathbf{x}}^{s-1} - \tilde{\mathbf{x}}^{s-2}\|^2}{\big\langle \tilde{\mathbf{x}}^{s-1} - \tilde{\mathbf{x}}^{s-2}, \nabla f(\tilde{\mathbf{x}}^{s-1}) - \nabla f(\tilde{\mathbf{x}}^{s-2}) \big\rangle}, \tag{7.6}$$

where θ_κ is a κ-dependent parameter to be specified later. Note that $\nabla f(\tilde{\mathbf{x}}^{s-1})$ and $\nabla f(\tilde{\mathbf{x}}^{s-2})$ are computed at the outer loops s and $s-1$, respectively; hence, the implementation overhead of BB step sizes only includes almost negligible memory to store $\tilde{\mathbf{x}}^{s-2}$ and $\nabla f(\tilde{\mathbf{x}}^{s-2})$.



(a) (b)

Figure 7.4: (a) Performance of BB-SVRG under different choices of m. (b) Performance of BB-SARAH with different averaging schemes. Both experiments use dataset *a9a* with $\kappa = 1,388$.

BB step sizes for SVRG with L-Avg have relied on $\theta_\kappa = m = \mathcal{O}(\kappa^2)$. Such a choice of parameters offers provable convergence at complexity $\mathcal{O}\big((n + \kappa^2)\ln\frac{1}{\epsilon}\big)$, but has not been effective in our simulations for two reasons: i) step size η^s depends on m, which means that tuning is still required for step sizes, and ii) the optimal m of $\mathcal{O}(\kappa)$ with best empirical performance significantly deviates from the theoretically suggested $\mathcal{O}(\kappa^2)$; see also Figure 7.4(a). Other BB-based variance reduction methods introduce extra parameters to be tuned in additional to m. This prompts us to design more practical BB methods; how to choose m with minimal tuning is also of major practical value.

7.3.2 Averaging for BB Step Sizes

We start with a fixed choice of m to theoretically investigate different types of averaging for the BB step sizes. The final "tune-free" implementation of SVRG and SARAH will rely on the analysis of this section.

Proposition 7.1. ***(BB-SVRG)*** *Under Assumptions 7.1 – 7.3, if we choose* $m = \mathcal{O}(\kappa^2)$ *and* $\theta_\kappa = \mathcal{O}(\kappa)$ *(but with* $\theta_\kappa > 4\kappa$*), then BB-SVRG with U-Avg and W-Avg can find* $\tilde{\mathbf{x}}^s$ *with* $\mathbb{E}\big[f(\tilde{\mathbf{x}}^s) - f(\mathbf{x}^*)\big] \leq \epsilon$ *using* $\mathcal{O}\big((n+\kappa^2)\ln\frac{1}{\epsilon}\big)$ *IFO calls.*

Similar to BB-SVRG, the ensuing result asserts that for BB-SARAH, W-Avg, U-Avg, and L-Avg have identical order of complexity.

Proposition 7.2. ***(BB-SARAH)*** *Under Assumptions 7.1 and 7.4, if we choose $m = \mathcal{O}(\kappa^2)$ and $\theta_\kappa = \mathcal{O}(\kappa)$, then BB-SARAH finds a solution with $\mathbb{E}\big[\|\nabla f(\tilde{\mathbf{x}}^s)\|^2\big] \leq \epsilon$ using $\mathcal{O}\big((n+\kappa^2)\ln\frac{1}{\epsilon}\big)$ IFO calls, when one of these conditions holds: i) either U-Avg with $\theta_\kappa > \kappa$; or ii) L-Avg with $\theta_\kappa > 3/2\kappa$; or, iii) W-Avg with $\theta_\kappa > \kappa$.*

The price paid for having automatically tuned step sizes is a worse dependence of the complexity on κ, compared with the bounds in Corollaries 7.1 and 7.2. The cause of the worse dependence on κ is that one has to choose a large m at the order of $\mathcal{O}(\kappa^2)$. However, such an automatic tuning of the step size comes almost as a "free lunch" when problem (1.3) is well conditioned, or in the big data regime, e.g., $\kappa^2 \approx n$ or $\kappa^2 \ll n$, since the dominant term in complexity is $\mathcal{O}(n\ln\frac{1}{\epsilon})$ for both SVRG and BB-SVRG. On the other hand, it is prudent to stress that with $\kappa^2 \gg n$, the BB step sizes slow down convergence.

Given the same order of complexity, the empirical performance of BB-SARAH with different averaging types is showcased in Figure 7.4(b) with the parameters chosen as in Proposition 7.2. It is observed that W-Avg converges most rapidly, while U-Avg outperforms L-Avg. This confirms our theoretical insight, that is, W-Avg and U-Avg are more suitable when m is chosen large enough.

7.3.3 Tune-Free Variance Reduction

Next, the ultimate format of the almost tune-free variance reduction is presented using SARAH as an example. We will discuss how to choose the iteration number of inner loops and averaging schemes for BB step sizes.

Adaptive inner loop length. It is observed that the BB step size can change over a wide range of values (see Section 7.7 for derivations),

$$\frac{1}{\theta_\kappa L} \leq \eta^s \leq \frac{1}{\theta_\kappa \mu}. \tag{7.7}$$

Given $\theta_\kappa = \mathcal{O}(\kappa)$, η^s can vary from $\mathcal{O}(\mu/L^2)$ to $\mathcal{O}(1/L)$. Such a wide range of η^s blocks the possibility to find a single m suitable for both small and large η^s at the same time. From a theoretical perspective, choosing $m = \mathcal{O}(\kappa^2)$ in both Propositions 7.1 and 7.2 is mainly for coping with the small step sizes $\eta^s = \mathcal{O}(1/(L\theta_\kappa))$. But such a choice is too pessimistic for large ones $\eta^s = \mathcal{O}(1/(\mu\theta_\kappa))$. In fact, choosing $m = \mathcal{O}(\kappa)$ for $\eta^s = \mathcal{O}(1/L)$ is good enough, as suggested by Corollaries 7.1 and 7.2. These observations motivate us to design an m^s that changes dynamically per outer loop s.

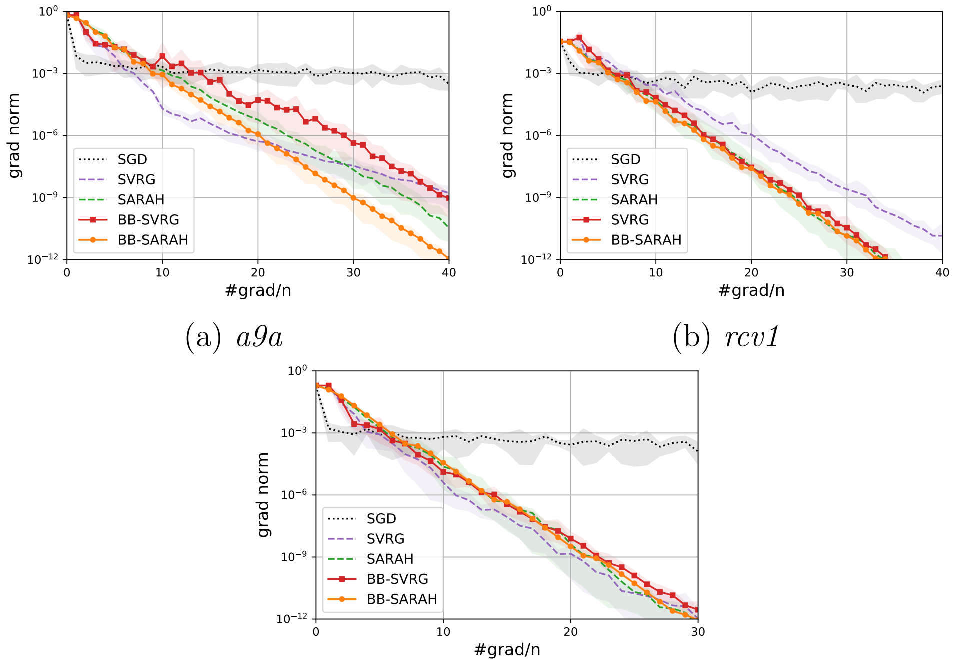
Reflecting on the convergence of SARAH, it is sufficient to set the inner loop length m^s according to the η^s used. To highlight the rationale behind our choice of m^s, let us consider BB-SARAH with U-Avg as an example that features convergence rate $\lambda^s = \frac{1}{\mu\eta^s m^s} + \frac{\eta^s L}{2-\eta^s L}$ [132]. Set $\theta_\kappa > \kappa$ as in Proposition 7.2 so that the second term of λ^s is always less than 1. With a large step size $\eta^s = \mathcal{O}(1/L)$, and by simply choosing $m^s = \mathcal{O}\big(1/(\mu\eta^s)\big)$, one can ensure a convergent iteration having e.g., $\lambda^s < 1$. With a small step size $\eta^s = \mathcal{O}\big(1/(\kappa L)\big)$ though, choosing $m^s = \mathcal{O}\big(1/(\mu\eta^s)\big)$ also leads to $\lambda^s < 1$. These considerations prompt us to adopt a time-varying inner loop length adjusted by η^s in (7.6) as

$$m^s = \frac{c}{\mu\eta^s}\,. \tag{7.8}$$

Such choices of η^s and m^s at first glance do not lead to a tune-free algorithm directly because one has to find an optimal θ_κ and c through tuning. Fortunately, there are simple choices for both c and θ_κ. In Propositions 7.1 and 7.2, the smallest selected θ_κ for SVRG and SARAH with different types of averaging turns out to be a reliable choice, while choosing $c = 1$ has been good enough throughout our numerical experiments. Although the selection of these parameters violates slightly the theoretical guarantee, its merits lie in the simplicity. And in our experiments, no divergence has been observed by these parameter selections.

Averaging schemes. As discussed in Section 7.2.4, W-Avg gains in performance when either m^s or η^s is large. Since m^s and η^s are inversely proportional (cf. (7.8)), it is clear that one of the two suffices to be large, and for this reason, we will rely on W-Avg for BB-SARAH.

Extensions regarding almost tune-free variance reduction for (non)convex problems can be found in our technical note [241].



(a) *a9a*

(b) *rcv1*

real-sim

Figure 7.5: experiments of BB-SVRG and BB-SARAH on different datasets.

7.4 Numerical Experiments

To assess performance, the proposed tune-free BB-SVRG and BB-SARAH are applied to binary classification via regularized logistic regression (cf. (7.5)) using the datasets *a9a*, *rcv1.binary*, and *real-sim* from LIBSVM[2]. Details regarding the datasets, the μ values used, and implementation details are deferred to Section 7.8.2.

For comparison, the selected benchmarks are SGD, SVRG with U-Avg, and SARAH with U-Avg. The step size for SGD is $\eta = 0.05/(L(n_e + 1))$ where n_e is the index of epochs. For SVRG and SARAH, we fix $m = 5\kappa$ and tune for the best step sizes. For BB-SVRG, we choose η^s and m^s as (7.8) with $\theta_\kappa = 4\kappa$ (as in Proposition 7.1) and $c = 1$. We choose $\theta_\kappa = \kappa$ (as in Proposition 7.2) and $c = 1$ for BB-SARAH. W-Avg is adopted for both BB-SVRG and BB-SARAH.

The results are showcased in Figure 7.5. We also tested BB-SVRG with parameters chosen as [140, Theorem 3.8]. However it only slightly outperforms SGD and hence is not plotted here (see the blue line in Figure 7.4(a)

[2] Online available at `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html`.

as a reference). On dataset *a9a*, BB-SARAH outperforms tuned SARAH. BB-SVRG is worse than SVRG initially but has similar performance around the 40th sample pass on the x-axis. On dataset *rcv1* however, BB-SARAH, BB-SVRG and SARAH have similar performance, improving over SVRG. On dataset *real-sim*, BB-SARAH performs almost identical to SARAH. BB-SVRG exhibits comparable performance with SVRG.

7.5 Properties of ES

7.5.1 Proof of Lemma 7.2

i) By definition, $\Phi_0(\mathbf{x})$ is μ_0-strongly convex, and by checking Hessian one can find that $\Phi_k(\mathbf{x})$ is μ_k-strongly convex with $\mu_k = (1-\delta_k)\mu_{k-1} + \delta_k \mu$.
ii) Clearly, $\mathbf{x}_0$ minimizes $\Phi_0(\mathbf{x})$, and $\Phi_k(\mathbf{x})$ is quadratic. Arguing by induction, suppose that $\mathbf{x}_{k-1}$ minimizes $\Phi_{k-1}(\mathbf{x})$, to obtain

$$\Phi_{k-1}(\mathbf{x}) = \Phi_{k-1}^* + \frac{\mu_{k-1}}{2}\|\mathbf{x}-\mathbf{x}_{k-1}\|^2 \quad \Rightarrow \quad \nabla\Phi_{k-1}(\mathbf{x}) = \mu_{k-1}(\mathbf{x}-\mathbf{x}_{k-1}).$$

By definition of $\Phi_k(\mathbf{x})$, we also have

$$\begin{aligned}\nabla\Phi_k(\mathbf{x}) &= (1-\delta_k)\nabla\Phi_{k-1}(\mathbf{x}) + \delta_k\mathbf{v}_{k-1} + \mu\delta_k(\mathbf{x}-\mathbf{x}_{k-1}) \\ &= (1-\delta_k)\mu_{k-1}(\mathbf{x}-\mathbf{x}_{k-1}) + \delta_k\mathbf{v}_{k-1} + \mu\delta_k(\mathbf{x}-\mathbf{x}_{k-1}). \qquad (7.9)\end{aligned}$$

Using $\mu_k = (1-\delta_k)\mu_{k-1} + \delta_k\mu$ and setting $\nabla\Phi_k(\mathbf{x}) = \mathbf{0}$, we find that $\mathbf{x}_k$ minimizes $\Phi_k(\mathbf{x})$ when $\delta_k = \eta\mu_k$.
iii) Since $\mathbf{x}_{k-1}$ minimizes $\Phi_{k-1}(\mathbf{x})$, using the definition of $\Phi_k(\mathbf{x})$ we can write

$$\Phi_k(\mathbf{x}_{k-1}) = (1-\delta_k)\Phi_{k-1}^* + \delta_k f(\mathbf{x}_{k-1}). \qquad (7.10)$$

On the other hand, we also have $\Phi_k(\mathbf{x}_{k-1}) = \Phi_k^* + \frac{\mu_k}{2}\|\mathbf{x}_{k-1}-\mathbf{x}_k\|^2$. Comparing this with (7.10) and using that $\mathbf{x}_k = \mathbf{x}_{k-1} - \eta\mathbf{v}_{k-1}$, completes the proof of this property.

7.5.2 Derivations of (7.3) and (7.4)

To verify (7.3), proceed as follows

$$\begin{aligned}
\mathbb{E}\big[\Phi_k(\mathbf{x})\big] &= (1-\delta)\mathbb{E}\big[\Phi_{k-1}(\mathbf{x})\big] \\
&\qquad + \delta\mathbb{E}\left[f(\mathbf{x}_{k-1}) + \langle \mathbf{v}_{k-1}, \mathbf{x}-\mathbf{x}_{k-1}\rangle + \frac{\mu}{2}\|\mathbf{x}-\mathbf{x}_{k-1}\|^2\right] \\
&= (1-\delta)\mathbb{E}\big[\Phi_{k-1}(\mathbf{x})\big] \\
&\qquad + \delta\mathbb{E}\left[f(\mathbf{x}_{k-1}) + \langle \nabla f(\mathbf{x}_{k-1}), \mathbf{x}-\mathbf{x}_{k-1}\rangle + \frac{\mu}{2}\|\mathbf{x}-\mathbf{x}_{k-1}\|^2\right] \\
&\le (1-\delta)\mathbb{E}\big[\Phi_{k-1}(\mathbf{x})\big] + \delta f(\mathbf{x}) \\
&\le (1-\delta)^k\big[\Phi_0(\mathbf{x}) - f(\mathbf{x})\big] + f(\mathbf{x}) \\
&\le (1-\delta)^k\big[\Phi_0(\mathbf{x}) - f(\mathbf{x}^*)\big] + f(\mathbf{x}). \qquad (7.11)
\end{aligned}$$

And in order to derive (7.4), follow the next steps

$$\begin{aligned}
\mathbb{E}\big[\Phi_k(\mathbf{x})\big] &= (1-\delta)\mathbb{E}\big[\Phi_{k-1}(\mathbf{x})\big] \\
&\qquad + \delta\mathbb{E}\left[f(\mathbf{x}_{k-1}) + \langle \mathbf{v}_{k-1}, \mathbf{x}-\mathbf{x}_{k-1}\rangle + \frac{\mu}{2}\|\mathbf{x}-\mathbf{x}_{k-1}\|^2\right] \\
&\le (1-\delta)\mathbb{E}\big[\Phi_{k-1}(\mathbf{x})\big] + \delta f(\mathbf{x}) + \delta\mathbb{E}\big[\langle \mathbf{v}_{k-1} - \nabla f(\mathbf{x}_{k-1}), \mathbf{x}-\mathbf{x}_{k-1}\rangle\big] \\
&\le (1-\delta)^k\big[\Phi_0(\mathbf{x}) - f(\mathbf{x})\big] + f(\mathbf{x}) \\
&\qquad + \underbrace{\delta\sum_{\tau=0}^{k-1}(1-\delta)^\tau\mathbb{E}\big[\langle \mathbf{v}_{k-1-\tau} - \nabla f(\mathbf{x}_{k-1-\tau}), \mathbf{x}-\mathbf{x}_{k-1-\tau}\rangle\big]}_{:=C;\ \ C\neq 0,\ \text{an extra term compared with SVRG}}.
\end{aligned}$$

7.5.3 A Key Lemma

The next lemma plays a major role in our analysis.

Lemma 7.4. *If we choose $\mu_0 = \mu$, $\delta_k = \mu_k\eta$, and $\Phi_0^* = f(\mathbf{x}_0)$ in the ES defined in (7.2), we then find that: i) $\mu_k = \mu, \forall k$; ii) $\delta := \delta_k = \mu\eta$; and iii) the following inequality holds*

$$\delta\sum_{\tau=1}^{k-1}(1-\delta)^{k-\tau-1}\big[f(\mathbf{x}_\tau) - f(\mathbf{x}^*)\big] + (1-\delta)^{k-1}\big[f(\mathbf{x}_0) - f(\mathbf{x}^*)\big]$$

$$\leq (1-\delta)^k\big[\Phi_0(\mathbf{x}^*) - f(\mathbf{x}^*)\big] + \frac{\mu\eta^2}{2}\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\|\mathbf{v}_{\tau-1}\|^2 + \delta\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\zeta_{\tau-1},$$

where $\zeta_{k-1} := \langle \mathbf{v}_{k-1} - \nabla f(\mathbf{x}_{k-1}), \mathbf{x}^* - \mathbf{x}_{k-1}\rangle$.

Proof. Since i) and ii) are straightforward to verify, we will prove iii). Using property iii) in Lemma 7.2, we find

$$\begin{aligned} f(\mathbf{x}_k) - \Phi_k^* &= f(\mathbf{x}_k) - (1-\delta_k)\Phi_{k-1}^* - \delta_k f(\mathbf{x}_{k-1}) + \frac{\mu_k\eta^2}{2}\|\mathbf{v}_{k-1}\|^2 \\ &= f(\mathbf{x}_k) - \Phi_{k-1}^* + \delta_k\big(\Phi_{k-1}^* - f(\mathbf{x}_{k-1})\big) + \frac{\mu_k\eta^2}{2}\|\mathbf{v}_{k-1}\|^2 \\ &= f(\mathbf{x}_k) - f(\mathbf{x}_{k-1}) + f(\mathbf{x}_{k-1}) - \Phi_{k-1}^* \\ &\qquad + \delta_k\big(\Phi_{k-1}^* - f(\mathbf{x}_{k-1})\big) + \frac{\mu_k\eta^2}{2}\|\mathbf{v}_{k-1}\|^2 \\ &= (1-\delta_k)\big[f(\mathbf{x}_{k-1}) - \Phi_{k-1}^*\big] + \xi_k, \end{aligned} \tag{7.12}$$

where ξ_k is defined as

$$\xi_k := f(\mathbf{x}_k) - f(\mathbf{x}_{k-1}) + \frac{\mu_k\eta^2}{2}\|\mathbf{v}_{k-1}\|^2.$$

Upon expanding $f(\mathbf{x}_{k-1}) - \Phi_{k-1}^*$ in (7.12), we have

$$\begin{aligned} f(\mathbf{x}_k) - \Phi_k^* &= (1-\delta_k)\big[f(\mathbf{x}_{k-1}) - \Phi_{k-1}^*\big] + \xi_k \\ &= \Big[\prod_{\tau=1}^{k}(1-\delta_\tau)\Big][f(\mathbf{x}_0) - \Phi_0^*] + \sum_{\tau=1}^{k}\xi_\tau\Big[\prod_{j=\tau+1}^{k}(1-\delta_j)\Big], \end{aligned} \tag{7.13}$$

from which we deduce that

$$\begin{aligned} \Phi_k^* \leq \Phi_k(\mathbf{x}^*) &= (1-\delta_k)\Phi_{k-1}(\mathbf{x}^*) \\ &\qquad + \delta_k\Big[f(\mathbf{x}_{k-1}) + \langle \mathbf{v}_{k-1}, \mathbf{x}^* - \mathbf{x}_{k-1}\rangle + \frac{\mu}{2}\|\mathbf{x}^* - \mathbf{x}_{k-1}\|^2\Big] \\ &\overset{(a)}{=} (1-\delta_k)\Phi_{k-1}(\mathbf{x}^*) \\ &\qquad + \delta_k\Big[f(\mathbf{x}_{k-1}) + \langle \nabla f(\mathbf{x}_{k-1}), \mathbf{x}^* - \mathbf{x}_{k-1}\rangle + \frac{\mu}{2}\|\mathbf{x}^* - \mathbf{x}_{k-1}\|^2 + \zeta_{k-1}\Big] \end{aligned}$$

$$\begin{aligned}
&\overset{(b)}{\leq} (1-\delta_k)\Phi_{k-1}(\mathbf{x}^*) + \delta_k f(\mathbf{x}^*) + \delta_k \zeta_{k-1} \\
&\leq \Big[\prod_{\tau=1}^{k}(1-\delta_\tau)\Big]\Phi_0(\mathbf{x}^*) + \sum_{\tau=1}^{k}\delta_\tau f(\mathbf{x}^*)\Big[\prod_{j=\tau+1}^{k}(1-\delta_j)\Big] \\
&\quad + \sum_{\tau=1}^{k}\delta_\tau\zeta_{\tau-1}\Big[\prod_{j=\tau+1}^{k}(1-\delta_j)\Big],
\end{aligned} \tag{7.14}$$

where in (a) the ζ_{k-1} is defined as

$$\zeta_{k-1} := \langle \mathbf{v}_{k-1} - \nabla f(\mathbf{x}_{k-1}), \mathbf{x}^* - \mathbf{x}_{k-1}\rangle;$$

and (b) follows from the strongly convexity of f. Then, using (7.13), we have

$$\begin{aligned}
f(\mathbf{x}_k) - f(\mathbf{x}^*) &= \Phi_k^* - f(\mathbf{x}^*) + \Big[\prod_{\tau=1}^{k}(1-\delta_\tau)\Big][f(\mathbf{x}_0) - \Phi_0^*] \\
&\quad + \sum_{\tau=1}^{k}\xi_\tau\Big[\prod_{j=\tau+1}^{k}(1-\delta_j)\Big] \\
&\overset{(c)}{\leq} \Big[\prod_{\tau=1}^{k}(1-\delta_\tau)\Big]\Phi_0(\mathbf{x}^*) + \sum_{\tau=1}^{k}\delta_\tau f(\mathbf{x}^*)\Big[\prod_{j=\tau+1}^{k}(1-\delta_j)\Big] \\
&\quad + \sum_{\tau=1}^{k}\delta_\tau\zeta_{\tau-1}\Big[\prod_{j=\tau+1}^{k}(1-\delta_j)\Big] - f(\mathbf{x}^*) \\
&\quad + \Big[\prod_{\tau=1}^{k}(1-\delta_\tau)\Big][f(\mathbf{x}_0) - \Phi_0^*] + \sum_{\tau=1}^{k}\xi_\tau\Big[\prod_{j=\tau+1}^{k}(1-\delta_j)\Big],
\end{aligned}$$

where (c) is due to (7.14). Choosing $\mu_0 = \mu$ (hence $\mu_k = \mu, \delta_k = \mu\eta := \delta,\ \forall k$) and $\Phi_0^* = f(\mathbf{x}_0)$, we arrive at

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq (1-\delta)^k\big[\Phi_0(\mathbf{x}^*) - f(\mathbf{x}^*)\big] + \sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\big(\xi_\tau + \delta\zeta_{\tau-1}\big). \tag{7.15}$$

Now consider that

$$\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\xi_\tau = \sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\Big[f(\mathbf{x}_\tau) - f(\mathbf{x}_{\tau-1}) + \frac{\mu\eta^2}{2}\|\mathbf{v}_{\tau-1}\|^2\Big]$$

$$
\begin{aligned}
&= f(\mathbf{x}_k) + \sum_{\tau=1}^{k-1}(1-\delta)^{k-\tau} f(\mathbf{x}_\tau) - \sum_{\tau=1}^{k-1}(1-\delta)^{k-\tau-1} f(\mathbf{x}_\tau) \\
&\qquad - (1-\delta)^{k-1} f(\mathbf{x}_0) + \frac{\mu\eta^2}{2}\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\|\mathbf{v}_{\tau-1}\|^2 \\
&= -\delta\sum_{\tau=1}^{k-1}(1-\delta)^{k-\tau-1} f(\mathbf{x}_\tau) + f(\mathbf{x}_k) - (1-\delta)^{k-1} f(\mathbf{x}_0) \\
&\qquad + \frac{\mu\eta^2}{2}\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\|\mathbf{v}_{\tau-1}\|^2. \qquad (7.16)
\end{aligned}
$$

Because $\delta\sum_{\tau=1}^{k-1}(1-\delta)^{k-\tau-1}+(1-\delta)^{k-1} = 1$, we can write $f(\mathbf{x}^*) = [\delta\sum_{\tau=1}^{k-1}(1-\delta)^{k-\tau-1} + (1-\delta)^{k-1}]f(\mathbf{x}^*)$. Using the latter, plugging (7.16) into (7.15), and eliminating $f(\mathbf{x}_k)$, we obtain

$$
\begin{aligned}
&\delta\sum_{\tau=1}^{k-1}(1-\delta)^{k-\tau-1}\big[f(\mathbf{x}_\tau) - f(\mathbf{x}^*)\big] + (1-\delta)^{k-1}\big[f(\mathbf{x}_0) - f(\mathbf{x}^*)\big] \\
&\leq (1-\delta)^k\big[\Phi_0(\mathbf{x}^*) - f(\mathbf{x}^*)\big] + \frac{\mu\eta^2}{2}\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\|\mathbf{v}_{\tau-1}\|^2 \\
&\qquad + \delta\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\zeta_{\tau-1}, \qquad (7.17)
\end{aligned}
$$

which completes the proof. □

7.6 Proofs for SVRG and SARAH

7.6.1 Proof for SVRG

Proof of Theorem 7.1

Proof. Since the choices of μ_0, Φ_0^*, and δ_k coincide with those in Lemma 7.4, we can directly apply Lemma 7.4 to find

$$
\delta\sum_{\tau=1}^{k-1}(1-\delta)^{k-\tau-1}\big[f(\mathbf{x}_\tau) - f(\mathbf{x}^*)\big] + (1-\delta)^{k-1}\big[f(\mathbf{x}_0) - f(\mathbf{x}^*)\big]
$$

$$\le (1-\delta)^k \big[\Phi_0(\mathbf{x}^*) - f(\mathbf{x}^*)\big] + \frac{\mu\eta^2}{2}\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\|\mathbf{v}_{\tau-1}\|^2$$
$$+ \delta\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\zeta_{\tau-1}, \tag{7.18}$$

where $\zeta_{k-1} := \langle \mathbf{v}_{k-1} - \nabla f(\mathbf{x}_{k-1}), \mathbf{x}^* - \mathbf{x}_{k-1}\rangle$. Upon defining the σ-algebra $\mathcal{F}_{k-1} = \sigma(i_0, i_1, \ldots, i_{k-1})$, and using that $\mathbf{v}_k$ is an unbiased estimate of $\nabla f(\mathbf{x}_k)$, it follows readily that

$$\mathbb{E}[\zeta_k|\mathcal{F}_{k-1}] = \mathbb{E}\big[\mathbf{v}_k - \nabla f(\mathbf{x}_k), \mathbf{x}^* - \mathbf{x}_k\rangle|\mathcal{F}_{k-1}\big] = 0,$$

which further implies

$$\mathbb{E}[\zeta_k] = 0. \tag{7.19}$$

Now taking expectation on both sides of (7.18) and using (7.19), we have

$$\delta\sum_{\tau=1}^{k-1}(1-\delta)^{k-\tau-1}\mathbb{E}\big[f(\mathbf{x}_\tau) - f(\mathbf{x}^*)\big] + (1-\delta)^{k-1}\mathbb{E}\big[f(\mathbf{x}_0) - f(\mathbf{x}^*)\big] \tag{7.20}$$
$$\le (1-\delta)^k\mathbb{E}\big[\Phi_0(\mathbf{x}^*) - f(\mathbf{x}^*)\big] + \frac{\mu\eta^2}{2}\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\mathbb{E}\big[\|\mathbf{v}_{\tau-1}\|^2\big]$$
$$\overset{(a)}{\le} (1-\delta)^k\mathbb{E}\big[\Phi_0(\mathbf{x}^*) - f(\mathbf{x}^*)\big]$$
$$+ 2\mu L\eta^2\sum_{\tau=0}^{k-1}(1-\delta)^{k-\tau-1}\mathbb{E}\big[f(\mathbf{x}_\tau) - f(\mathbf{x}^*) + f(\mathbf{x}_0) - f(\mathbf{x}^*)\big]$$
$$\overset{(b)}{\le} (1-\delta)^k\mathbb{E}\big[\Phi_0(\mathbf{x}^*) - f(\mathbf{x}^*)\big]$$
$$+ 2\mu L\eta^2\sum_{\tau=0}^{k-1}(1-\delta)^{k-\tau-1}\mathbb{E}\big[f(\mathbf{x}_\tau) - f(\mathbf{x}^*)\big] + \frac{2\mu L\eta^2}{\delta}\mathbb{E}\big[f(\mathbf{x}_0) - f(\mathbf{x}^*)\big],$$

where in (a) we used Lemma 7.1 to $\mathbb{E}[\|\mathbf{v}_{\tau-1}\|^2]$, and (b) holds because $\sum_{\tau=0}^{k-1}(1-\delta)^{k-\tau-1} \le 1/\delta$. Note that we can use $\Phi_0(\mathbf{x}^*) = f(\mathbf{x}_0) + \frac{\mu}{2}\|\mathbf{x}_0 - \mathbf{x}^*\|^2$ together with $(1-\delta)^{k-1} > (1-\delta)^k$, to eliminate $(1-\delta)^{k-1}\mathbb{E}[f(\mathbf{x}_0) - f(\mathbf{x}^*)]$

on the LHS of (7.20). Rearranging the terms, we arrive at

$$\begin{aligned}
&(\delta - 2\mu L\eta^2)\sum_{\tau=1}^{k-1}(1-\delta)^{k-\tau-1}\mathbb{E}\big[f(\mathbf{x}_\tau) - f(\mathbf{x}^*)\big] \\
&\le \frac{\mu}{2}(1-\delta)^k\mathbb{E}\big[\|\mathbf{x}_0 - \mathbf{x}^*\|^2\big] + 2\mu L\eta^2(1-\delta)^{k-1}\mathbb{E}\big[f(\mathbf{x}_0) - f(\mathbf{x}^*)\big] \\
&\quad + \frac{2\mu L\eta^2}{\delta}\mathbb{E}\big[f(\mathbf{x}_0) - f(\mathbf{x}^*)\big] \\
&\le \left[(1-\delta)^k + 2\mu L\eta^2(1-\delta)^{k-1} + \frac{2\mu L\eta^2}{\delta}\right]\mathbb{E}\big[f(\mathbf{x}_0) - f(\mathbf{x}^*)\big], \qquad (7.21)
\end{aligned}$$

where the last inequality is due to $\frac{\mu}{2}\|\mathbf{x} - \mathbf{x}^*\| \le f(\mathbf{x}) - f(\mathbf{x}^*)$. Now choosing $\eta < 1/2L$ so that $\delta - 2\mu L\eta^2 > 0$, we have

$$\begin{aligned}
&\sum_{\tau=1}^{k-1}(1-\delta)^{k-\tau-1}\mathbb{E}\big[f(\mathbf{x}_\tau) - f(\mathbf{x}^*)\big] \\
&\le \left[\frac{(1-\delta)^k}{\delta - 2\mu L\eta^2} + \frac{2\mu L\eta^2(1-\delta)^{k-1}}{\delta - 2\mu L\eta^2} + \frac{2\mu L\eta^2}{\delta(\delta - 2\mu L\eta^2)}\right]\mathbb{E}\big[f(\mathbf{x}_0) - f(\mathbf{x}^*)\big].
\end{aligned}$$

With $p_0 = p_m = 0$, and $p_k = (1-\delta)^{m-k-1}/q, k = 1, 2, \ldots, m-1$, where $q = [1 - (1-\delta)^{m-1}]/\delta$ (with $\delta = \mu\eta$), we find

$$\begin{aligned}
&\mathbb{E}\big[f(\tilde{\mathbf{x}}^s) - f(\mathbf{x}^*)\big] = \sum_{\tau=1}^{m-1}\frac{(1-\delta)^{m-\tau-1}}{q}\mathbb{E}\big[f(\mathbf{x}_\tau) - f(\mathbf{x}^*)\big] \\
&\le \frac{1}{q}\left[\frac{(1-\delta)^m}{\delta - 2\mu L\eta^2} + \frac{2\mu L\eta^2(1-\delta)^{m-1}}{\delta - 2\mu L\eta^2} + \frac{2\mu L\eta^2}{\delta(\delta - 2\mu L\eta^2)}\right]\mathbb{E}\big[f(\tilde{\mathbf{x}}^{s-1}) - f(\mathbf{x}^*)\big] \\
&= \underbrace{\frac{1}{1-(1-\mu\eta)^{m-1}}\left[\frac{(1-\mu\eta)^m}{1-2\eta L} + \frac{2\mu L\eta^2(1-\mu\eta)^{m-1}}{1-2L\eta} + \frac{2L\eta}{1-2L\eta}\right]}_{:=\lambda^{\texttt{SVRG}}} \\
&\qquad \times \mathbb{E}\big[f(\tilde{\mathbf{x}}^{s-1}) - f(\mathbf{x}^*)\big]. \qquad (7.22)
\end{aligned}$$

Thus, so long as we choose a large enough m and $\eta < 1/(4L)$, we have $\lambda^{\texttt{SVRG}} < 1$, that is, SVRG converges linearly. □

Proof of Corollary 7.1

Proof. Choose $\eta = 1/(8L)$ and $m = \frac{3}{\mu\eta} + 1 = 24\kappa + 1 \ge 25$. We have that

$$(1-\mu\eta)^{\frac{1}{\mu\eta}} \le 0.4 \quad \Rightarrow \quad (1-\mu\eta)^m \le (0.4)^3$$

(Actually $(1-\mu\eta)^{\frac{1}{\mu\eta}} \approx 1/e$ when $\mu\eta$ is small enough). Using the value of η and m, it can be verified that $\lambda^{\texttt{SVRG}} \leq 0.5$. This implies that $\mathcal{O}\big(\ln\frac{1}{\epsilon}\big)$ outer loops are needed for an ϵ-accurate solution. And since $m = \mathcal{O}(\kappa)$, the overall complexity is $\mathcal{O}\big((n+\kappa)\ln\frac{1}{\epsilon}\big)$. □

7.6.2 Proofs for SARAH

Proof of Lemma 7.3

Proof. Let $\mathcal{F}_{k-1} = \sigma(i_1, i_2, \dots, i_{k-1})$, then for any $\mathbf{x}$ we have

$$\begin{aligned}
&\mathbb{E}\big[\langle \mathbf{v}_k - \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k\rangle | \mathcal{F}_{k-1}\big] \\
&= \mathbb{E}\big[\langle \nabla f_{i_k}(\mathbf{x}_k) - \nabla f_{i_k}(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1} - \nabla f(\mathbf{x}_k),\ \mathbf{x} - \mathbf{x}_k\rangle | \mathcal{F}_{k-1}\big] \\
&= \langle \mathbf{v}_{k-1} - \nabla f(\mathbf{x}_{k-1}),\ \mathbf{x} - \mathbf{x}_k\rangle \\
&= \langle \mathbf{v}_{k-1} - \nabla f(\mathbf{x}_{k-1}),\ \mathbf{x} - \mathbf{x}_{k-1} + \mathbf{x}_{k-1} - \mathbf{x}_k\rangle \\
&= \langle \mathbf{v}_{k-1} - \nabla f(\mathbf{x}_{k-1}),\ \mathbf{x} - \mathbf{x}_{k-1}\rangle + \eta\langle \mathbf{v}_{k-1} - \nabla f(\mathbf{x}_{k-1}),\ \mathbf{v}_{k-1}\rangle \\
&= \langle \mathbf{v}_{k-1} - \nabla f(\mathbf{x}_{k-1}),\ \mathbf{x} - \mathbf{x}_{k-1}\rangle \\
&\qquad + \frac{\eta}{2}\Big[\|\mathbf{v}_{k-1}\|^2 + \|\mathbf{v}_{k-1} - \nabla f(\mathbf{x}_{k-1})\|^2 - \|\nabla f(\mathbf{x}_{k-1})\|^2\Big],
\end{aligned}$$

where the last equation is because $2\langle \mathbf{a}, \mathbf{b}\rangle = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - \|\mathbf{a}-\mathbf{b}\|^2$. Since $\mathbf{v}_0 = \nabla f(\mathbf{x}_0)$, we have $\langle \mathbf{v}_0 - \nabla f(\mathbf{x}_0), \mathbf{x} - \mathbf{x}_0\rangle = 0$. The proof is completed after taking expectation and unrolling $\langle \mathbf{v}_{k-1} - \nabla f(\mathbf{x}_{k-1}), \mathbf{x} - \mathbf{x}_{k-1}\rangle$. □

In order to prove Theorem 7.2, we need to borrow the following result from [132].

Lemma 7.5. *[132, Theorem 1b] If Assumptions 7.1 and 7.4 hold, with $\eta \leq 2/(\mu + L)$, SARAH guarantees*

$$\mathbb{E}\big[\|\mathbf{v}_k\|^2\big] \leq \bigg(1 - \frac{2\eta L}{1+\kappa}\bigg)^k \mathbb{E}\big[\|\nabla f(\mathbf{x}_0)\|^2\big].$$

Proof of Theorem 7.2.

Proof. With the choices of μ_0, Φ_0^* and δ_k as in Lemma 7.4, we can directly apply Lemma 7.4 to confirm that

$$
\begin{aligned}
&(1-\delta)^{k-1}\big[f(\mathbf{x}_0)-f(\mathbf{x}^*)\big]+\delta\sum_{\tau=1}^{k-1}(1-\delta)^{k-\tau-1}\big[f(\mathbf{x}_\tau)-f(\mathbf{x}^*)\big]\\
&\le(1-\delta)^k\big[\Phi_0(\mathbf{x}^*)-f(\mathbf{x}^*)\big]+\frac{\mu\eta^2}{2}\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\|\mathbf{v}_{\tau-1}\|^2\\
&\qquad+\sum_{\tau=1}^{k}\delta(1-\delta)^{k-\tau}\langle\mathbf{v}_{\tau-1}-\nabla f(\mathbf{x}_{\tau-1}),\mathbf{x}^*-\mathbf{x}_{\tau-1}\rangle\\
&=(1-\delta)^k\big[\Phi_0(\mathbf{x}^*)-f(\mathbf{x}^*)\big]+\frac{\mu\eta^2}{2}\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\|\mathbf{v}_{\tau-1}\|^2\\
&\qquad+\sum_{\tau=2}^{k}\delta(1-\delta)^{k-\tau}\langle\mathbf{v}_{\tau-1}-\nabla f(\mathbf{x}_{\tau-1}),\mathbf{x}^*-\mathbf{x}_{\tau-1}\rangle,
\end{aligned}
$$

where the last equation holds because $\mathbf{v}_0=\nabla f(\mathbf{x}_0)$. Since $\Phi_0(\mathbf{x}^*)=f(\mathbf{x}_0)+\frac{\mu}{2}\|\mathbf{x}_0-\mathbf{x}^*\|^2\le f(\mathbf{x}_0)+\frac{1}{2\mu}\|\nabla f(\mathbf{x}_0)\|^2$ and $(1-\delta)^{k-1}>(1-\delta)^k$, we can eliminate $(1-\delta)^{k-1}\mathbb{E}[f(\mathbf{x}_0)-f(\mathbf{x}^*)]$ on the LHS, to obtain the inequality

$$
\begin{aligned}
&\delta\sum_{\tau=1}^{k-1}(1-\delta)^{k-\tau-1}\big[f(\mathbf{x}_\tau)-f(\mathbf{x}^*)\big]\\
&\le\frac{(1-\delta)^k}{2\mu}\|\nabla f(\mathbf{x}_0)\|^2+\frac{\mu\eta^2}{2}\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\|\mathbf{v}_{\tau-1}\|^2\\
&\qquad+\sum_{\tau=2}^{k}\delta(1-\delta)^{k-\tau}\langle\mathbf{v}_{\tau-1}-\nabla f(\mathbf{x}_{\tau-1}),\mathbf{x}^*-\mathbf{x}_{\tau-1}\rangle.
\end{aligned}
$$

Taking expectation on both sides, we arrive at

$$
\begin{aligned}
0&\le\delta\sum_{\tau=1}^{k-1}(1-\delta)^{k-\tau-1}\mathbb{E}\big[f(\mathbf{x}_\tau)-f(\mathbf{x}^*)\big] \qquad (7.23)\\
&\le\frac{(1-\delta)^k}{2\mu}\mathbb{E}\big[\|\nabla f(\mathbf{x}_0)\|^2\big]+\frac{\mu\eta^2}{2}\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\mathbb{E}\big[\|\mathbf{v}_{\tau-1}\|^2\big]\\
&\qquad+\sum_{\tau=2}^{k}\delta(1-\delta)^{k-\tau}\mathbb{E}\big[\langle\mathbf{v}_{\tau-1}-\nabla f(\mathbf{x}_{\tau-1}),\mathbf{x}^*-\mathbf{x}_{\tau-1}\rangle\big]
\end{aligned}
$$

$$\begin{aligned}
&= \frac{(1-\delta)^k}{2\mu}\mathbb{E}\big[\|\nabla f(\mathbf{x}_0)\|^2\big] + \frac{\mu\eta^2}{2}\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\mathbb{E}\big[\|\mathbf{v}_{\tau-1}\|^2\big] \\
&\quad + \sum_{\tau=1}^{k-1}\delta(1-\delta)^{k-1-\tau}\mathbb{E}\big[\langle \mathbf{v}_\tau - \nabla f(\mathbf{x}_\tau), \mathbf{x}^* - \mathbf{x}_\tau\rangle\big] \\
&\leq \frac{(1-\delta)^k}{2\mu}\mathbb{E}\big[\|\nabla f(\mathbf{x}_0)\|^2\big] + \frac{\mu\eta^2}{2}\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\mathbb{E}\big[\|\mathbf{v}_{\tau-1}\|^2\big] \\
&\quad + \frac{\delta\eta}{2}\sum_{\tau=1}^{k-1}(1-\delta)^{k-1-\tau}\sum_{j=0}^{\tau-1}\mathbb{E}\Big[\|\mathbf{v}_j - \nabla f(\mathbf{x}_j)\|^2 + \|\mathbf{v}_j\|^2 - \|\nabla f(\mathbf{x}_j)\|^2\Big],
\end{aligned}$$

where for the last inequality, we used Lemma 7.3. Changing the summation order in the last term of the RHS of (7.23), yields

$$\begin{aligned}
&\frac{\delta\eta}{2}\sum_{\tau=1}^{k-1}(1-\delta)^{k-1-\tau}\sum_{j=0}^{\tau-1}\mathbb{E}\Big[\|\mathbf{v}_j - \nabla f(\mathbf{x}_j)\|^2 + \|\mathbf{v}_j\|^2 - \|\nabla f(\mathbf{x}_j)\|^2\Big] \\
&= \frac{\delta\eta}{2}\sum_{\tau=0}^{k-2}\mathbb{E}\Big[\|\mathbf{v}_\tau - \nabla f(\mathbf{x}_\tau)\|^2 + \|\mathbf{v}_\tau\|^2 - \|\nabla f(\mathbf{x}_\tau)\|^2\Big]\Bigg[\sum_{j=0}^{k-\tau-2}(1-\delta)^\tau\Bigg] \\
&\leq \frac{\eta}{2}\sum_{\tau=0}^{k-2}\Big(\mathbb{E}\big[\|\mathbf{v}_\tau - \nabla f(\mathbf{x}_\tau)\|^2\big] + \mathbb{E}\big[\|\mathbf{v}_\tau\|^2\big]\Big) \\
&\quad - \frac{\eta}{2}\sum_{\tau=0}^{k-2}\big(1-(1-\delta)^{k-\tau-1}\big)\mathbb{E}\big[\|\nabla f(\mathbf{x}_\tau)\|^2\big]. \qquad (7.24)
\end{aligned}$$

Now plugging (7.24) into (7.23), and rearranging the terms, we find

$$\begin{aligned}
&\frac{\eta}{2}\sum_{\tau=0}^{k-2}\big(1-(1-\delta)^{k-1-\tau}\big)\mathbb{E}\big[\|\nabla f(\mathbf{x}_\tau)\|^2\big] \\
&\leq \frac{(1-\delta)^k}{2\mu}\mathbb{E}\big[\|\nabla f(\mathbf{x}_0)\|^2\big] + \frac{\mu\eta^2}{2}\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\mathbb{E}\big[\|\mathbf{v}_{\tau-1}\|^2\big] \\
&\quad + \frac{\eta}{2}\sum_{\tau=0}^{k-2}\Big(\mathbb{E}\big[\|\mathbf{v}_\tau - \nabla f(\mathbf{x}_\tau)\|^2\big] + \mathbb{E}\big[\|\mathbf{v}_\tau\|^2\big]\Big).
\end{aligned}$$

Dividing both sides by $\eta/2$ (and recalling that $\delta = \mu\eta$), we arrive at

$$\sum_{\tau=0}^{k-2}\big(1-(1-\delta)^{k-\tau-1}\big)\mathbb{E}\big[\|\nabla f(\mathbf{x}_\tau)\|^2\big] \qquad (7.25)$$

$$
\begin{aligned}
&\leq \frac{(1-\delta)^k}{\mu\eta}\mathbb{E}\big[\|\nabla f(\mathbf{x}_0)\|^2\big] + \delta\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\mathbb{E}\big[\|\mathbf{v}_{\tau-1}\|^2\big] \\
&\qquad + \sum_{\tau=0}^{k-2}\Big(\mathbb{E}\big[\|\mathbf{v}_\tau - \nabla f(\mathbf{x}_\tau)\|^2\big] + \mathbb{E}\big[\|\mathbf{v}_\tau\|^2\big]\Big) \\
&\overset{(a)}{\leq} \frac{(1-\delta)^k}{\mu\eta}\mathbb{E}\big[\|\nabla f(\mathbf{x}_0)\|^2\big] + \delta\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\mathbb{E}\big[\|\mathbf{v}_{\tau-1}\|^2\big] \\
&\qquad + \frac{\eta L(k-1)}{2-\eta L}\mathbb{E}\big[\|\nabla f(\mathbf{x}_0)\|^2\big] + \frac{2-2\eta L}{2-\eta L}\sum_{\tau=0}^{k-2}\mathbb{E}\big[\|\mathbf{v}_\tau\|^2\big] \\
&\overset{(b)}{\leq} \frac{(1-\delta)^k}{\mu\eta}\mathbb{E}\big[\|\nabla f(\mathbf{x}_0)\|^2\big] + \delta\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\mathbb{E}\big[\|\mathbf{v}_{\tau-1}\|^2\big] \\
&\qquad + \frac{\eta L(k-1)}{2-\eta L}\mathbb{E}\big[\|\nabla f(\mathbf{x}_0)\|^2\big] + \frac{2-2\eta L}{2-\eta L}\frac{1+\kappa}{2\eta L}\mathbb{E}\big[\|\nabla f(\mathbf{x}_0)\|^2\big] \\
&\overset{(c)}{\leq} \frac{(1-\delta)^k}{\mu\eta}\mathbb{E}\big[\|\nabla f(\mathbf{x}_0)\|^2\big] + \bigg[(1-\delta)^k - \Big(1-\frac{2\eta L}{1+\kappa}\Big)^k\bigg]\frac{L+\mu}{L-\mu}\mathbb{E}\big[\|\nabla f(\mathbf{x}_0)\|^2\big] \\
&\qquad + \frac{\eta L(k-1)}{2-\eta L}\mathbb{E}\big[\|\nabla f(\mathbf{x}_0)\|^2\big] + \frac{2-2\eta L}{2-\eta L}\frac{1+\kappa}{2L\eta}\mathbb{E}\big[\|\nabla f(\mathbf{x}_0)\|^2\big],
\end{aligned}
$$

where in (a) we applied Lemma 7.1 to deal with $\mathbb{E}[\|\nabla f(\mathbf{x}_\tau) - \mathbf{v}_\tau\|^2]$; in (b) we chose $\eta < 1/L$ and used Lemma 7.5 to handle $\mathbb{E}[\|\mathbf{v}_\tau\|^2]$ in the last term; and the derivation of (c) is as follows. First, notice that $2\eta L/(1+\kappa) > \mu\eta = \delta$, which implies that $1-\delta > 1-[2\eta L/(1+\kappa)]$. Then, leveraging Lemma 7.5, we have

$$
\begin{aligned}
&\delta\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\mathbb{E}\big[\|\mathbf{v}_{\tau-1}\|^2\big] \leq \delta\sum_{\tau=1}^{k}(1-\delta)^{k-\tau}\Big(1-\frac{2\eta L}{1+\kappa}\Big)^{\tau-1}\mathbb{E}\big[\|\nabla f(\mathbf{x}_0)\|^2\big] \\
&= \bigg[(1-\delta)^k - \Big(1-\frac{2\eta L}{1+\kappa}\Big)^k\bigg]\frac{L+\mu}{L-\mu}\mathbb{E}\big[\|\nabla f(\mathbf{x}_0)\|^2\big].
\end{aligned}
$$

To proceed, define

$$
\begin{aligned}
c := \sum_{\tau=0}^{m-2}\big(1-(1-\delta)^{m-\tau-1}\big) &= (m-1) - \frac{(1-\delta)-(1-\delta)^m}{\delta} \\
&= m - \frac{1}{\delta} + \frac{(1-\delta)^m}{\delta},
\end{aligned}
$$

and select m large enough so that $c > 0$. Upon setting $p_k = (1-(1-$

$\delta)^{m-k-1})/c, \forall k = 0, 1, \ldots, m-2$, and $p_{m-1} = p_m = 0$, we have

$$\mathbb{E}\big[\|\nabla f(\tilde{\mathbf{x}}^s)\|\big] = \frac{1}{c}\sum_{\tau=0}^{m-2}\big(1-(1-\delta)^{m-\tau-1}\big)\mathbb{E}\big[\|\nabla f(\mathbf{x}_\tau)\|^2\big]$$
$$\leq \underbrace{\bigg[\frac{(1-\delta)^m}{c\mu\eta} + \bigg((1-\delta)^m - \Big(1-\frac{2\eta L}{1+\kappa}\Big)^m\bigg)\frac{L+\mu}{c(L-\mu)}}$$
$$\underbrace{+\frac{\eta L(m-1)}{c(2-\eta L)} + \frac{2-2\eta L}{2-\eta L}\frac{1+\kappa}{2c\eta L}\bigg]}_{:=\lambda^{\texttt{SARAH}}}\mathbb{E}\big[\|\nabla f(\tilde{\mathbf{x}}^{s-1})\|^2\big].$$

Selecting $\eta < 1/L$ and m large enough to let $\lambda^{\texttt{SARAH}} < 1$ establishes SARAH's linear convergence. For example, choosing $\eta = 1/(2L)$ and $m = 5\kappa$, we have $\lambda^{\texttt{SARAH}} \approx 0.8$. □

Proof of Corollary 7.2

Proof. If we choose $\eta = 1/(2L)$ and $m = 6\kappa = 3/(\mu\eta)$, we have $\delta = 1/(2\kappa)$ and $c \geq 4\kappa$, which implies that

$$(1-\mu\eta)^{\frac{1}{\mu\eta}} \leq 0.4$$

(actually $(1-\mu\eta)^{\frac{1}{\mu\eta}} \approx 1/e$ when $\mu\eta$ small enough). Using the value of η and m, it can be verified that $\lambda^{\texttt{SVRG}} \leq 0.75$. This implies that $\mathcal{O}\big(\ln\frac{1}{\epsilon}\big)$ outer loops are needed for an ϵ-accurate solution. Since $m = \mathcal{O}(\kappa)$, the overall complexity is $\mathcal{O}\big((n+\kappa)\ln\frac{1}{\epsilon}\big)$. □

7.7 Proofs for BB-SVRG and BB-SARAH

Derivation of (7.7)**:** It is clear that

$$\eta^s = \frac{1}{\theta_\kappa}\frac{\|\tilde{\mathbf{x}}^{s-1}-\tilde{\mathbf{x}}^{s-2}\|^2}{\big\langle\tilde{\mathbf{x}}^{s-1}-\tilde{\mathbf{x}}^{s-2}, \nabla f(\tilde{\mathbf{x}}^{s-1})-\nabla f(\tilde{\mathbf{x}}^{s-2})\big\rangle} \leq \frac{1}{\theta_\kappa}\frac{\|\tilde{\mathbf{x}}^{s-1}-\tilde{\mathbf{x}}^{s-2}\|^2}{\mu\|\tilde{\mathbf{x}}^{s-1}-\tilde{\mathbf{x}}^{s-2}\|^2} = \frac{1}{\theta_\kappa\mu},$$

where the inequality follows since under Assumption 7.3 (or 7.4) $\langle\nabla f(\mathbf{x}) - \nabla f(\mathbf{y}), \mathbf{x}-\mathbf{y}\rangle \geq \mu\|\mathbf{x}-\mathbf{y}\|^2$ [125, Theorem 2.1.9]. On the other hand, we have

$$\eta^s \geq \frac{1}{\theta_\kappa} \frac{\|\tilde{\mathbf{x}}^{s-1} - \tilde{\mathbf{x}}^{s-2}\|^2}{\|\tilde{\mathbf{x}}^{s-1} - \tilde{\mathbf{x}}^{s-2}\| \big\|\nabla f(\tilde{\mathbf{x}}^{s-1}) - \nabla f(\tilde{\mathbf{x}}^{s-2})\big\|} \geq \frac{1}{\theta_\kappa L},$$

where the first inequality follows from the Cauchy-Schwarz inequality, and the second inequality is due to Assumption 7.1.

7.7.1 Proof for Proposition 7.1

For BB-SVRG, the step size η^s changes across different inner loops. Since η^s influences convergence, we will use λ^s to denote the convergence rate of the inner loop s, that is, $\mathbb{E}[f(\tilde{\mathbf{x}}^s) - f(\mathbf{x}^*)] \leq \lambda^s \mathbb{E}[f(\tilde{\mathbf{x}}^{s-1}) - f(\mathbf{x}^*)]$.

BB-SVRG with U-Avg:

Proof. From [128], we have the convergence rate is

$$\lambda^s = \frac{1}{\mu\eta^s(1 - 2\eta^s L)m} + \frac{2\eta^s L}{1 - 2\eta^s L} \overset{(a)}{\leq} \frac{\kappa\theta_\kappa}{m(1 - 2\kappa/\theta_\kappa)} + \frac{2\kappa/\theta_\kappa}{1 - 2\kappa/\theta_\kappa},$$

where (a) is due to (7.7). Hence, by choosing $\theta_\kappa > 4\kappa$ with $\theta_\kappa = \mathcal{O}(\kappa)$ and $m = \mathcal{O}(\kappa^2)$ such that $\lambda^s < 1$, and using similar arguments as in the proof of Corollary 7.1, one can readily verify that the complexity is $\mathcal{O}\big((n + \kappa^2)\ln\frac{1}{\epsilon}\big)$. □

BB-SVRG with W-Avg:

Proof. It follows from Theorem 7.1 and (7.7) that the convergence rate satisfies

$$\begin{aligned}
\lambda^s &= \frac{1}{1 - (1 - \mu\eta^s)^{m-1}} \left[\frac{(1 - \mu\eta^s)^m}{1 - 2\eta^s L} + \frac{2\mu L(\eta^s)^2(1 - \mu\eta)^{m-1}}{1 - 2L\eta^s} + \frac{2L\eta^s}{1 - 2L\eta^s} \right] \\
&\leq \frac{1}{1 - \big(1 - \frac{1}{\kappa\theta_\kappa}\big)^{m-1}} \left[\frac{\big(1 - \frac{1}{\kappa\theta_\kappa}\big)^m}{1 - 2\kappa/\theta_\kappa} + \frac{\frac{2\kappa}{(\theta_\kappa)^2}\big(1 - \frac{1}{\kappa\theta_\kappa}\big)^{m-1}}{1 - 2\kappa/\theta_\kappa} + \frac{2\kappa/\theta_\kappa}{1 - 2\kappa/\theta_\kappa} \right],
\end{aligned}$$

where the inequality is due to (7.7). Hence, by choosing $\theta_\kappa > 4\kappa$ with $\theta_\kappa = \mathcal{O}(\kappa)$ and $m = \mathcal{O}(\kappa^2)$ so that $\lambda^s < 1$, and using similar arguments as in the proof of Corollary 7.1, one can establish that the complexity is $\mathcal{O}\big((n + \kappa^2)\ln\frac{1}{\epsilon}\big)$. □

7.7.2 Proof for Proposition 7.2

Also for BB-SARAH, the step size η^s changes across different inner loops. Since here too η^s affects convergence, we will use λ^s to denote the convergence rate of the inner loop s; that is, $\mathbb{E}[\|f(\tilde{\mathbf{x}}^s)\|^2] \leq \lambda^s \mathbb{E}[\|f(\tilde{\mathbf{x}}^{s-1})\|^2]$.

BB-SARAH with U-Avg:

Proof. We have from [132] that the convergence rate is

$$\lambda^s = \frac{1}{\mu \eta^s m} + \frac{\eta^s L}{2 - \eta^s L} \overset{(a)}{\leq} \frac{\kappa \theta_\kappa}{m} + \frac{\kappa/\theta_\kappa}{2 - \kappa/\theta_\kappa},$$

where (a) is due to (7.7). Hence, by choosing $\theta_\kappa > \kappa$ with $\theta_\kappa = \mathcal{O}(\kappa)$ and $m = \mathcal{O}(\kappa^2)$ so that $\lambda^s < 1$, and using arguments similar to those in the proof of Corollary 7.2, one can establish that the complexity is $\mathcal{O}\big((n+\kappa^2)\ln\frac{1}{\epsilon}\big)$. □

BB-SARAH with L-Avg:

Proof. Since the derivation in [136] relies on Assumption 7.3, we will first establish the convergence rate under Assumption 7.4. The proof proceeds along the lines of [136], except for the use of Lemma 7.5 to bound $\mathbb{E}[\|\mathbf{v}_t^s\|]^2$. After a simple derivation, one can have the convergence rate

$$\lambda^s = \frac{2\eta^s L}{2 - \eta^s L} + 2(1 + \eta^s L)\left(1 - \frac{2\eta^s L}{1+\kappa}\right)^m.$$

Then using (7.7) to upper bound λ^s, we have

$$\lambda^s \leq \frac{2\kappa/\theta_\kappa}{2 - \kappa/\theta_\kappa} + 2(1 + \kappa/\theta_\kappa)\left(1 - \frac{2}{(1+\kappa)\theta_\kappa}\right)^m.$$

Hence, by choosing $\theta_\kappa > 3\kappa/2$ with $\theta_\kappa = \mathcal{O}(\kappa)$ and $m = \mathcal{O}(\kappa^2)$ so that $\lambda^s < 1$, and using arguments similar to those in the proof of Corollary 7.2, one can verify that the complexity is $\mathcal{O}\big((n+\kappa^2)\ln\frac{1}{\epsilon}\big)$. □

BB-SARAH with W-Avg:

Proof. From Theorem 7.2, the convergence rate is

$$\lambda^s = \frac{(1-\mu\eta^s)^m}{c\mu\eta^s} + \left[(1-\mu\eta^s)^m - \left(1-\frac{2\eta^s L}{1+\kappa}\right)^m\right]\frac{L+\mu}{c(L-\mu)} \\ + \frac{\eta^s L(m-1)}{c(2-\eta^s L)} + \frac{2-2\eta^s L}{2-\eta^s L}\frac{1+\kappa}{2c\eta^s L}$$

$$\begin{aligned}\le \frac{\kappa\theta_\kappa\big(1-\frac{1}{\kappa\theta_\kappa}\big)^m}{c} &+ \big(1-\frac{1}{\kappa\theta_\kappa}\big)^m \frac{L+\mu}{c(L-\mu)}\\ &+ \frac{(m-1)\kappa/\theta_\kappa}{c(2-\kappa/\theta_\kappa)} + \frac{2}{2-\kappa/\theta_\kappa}\frac{(1+\kappa)\theta_\kappa}{2c},\end{aligned}$$

where $c = m - \frac{1}{\mu\eta^s} + \frac{(1-\mu\eta^s)^m}{\mu\eta^s} \ge m - \frac{1}{\mu\eta^s} \ge m - \kappa\theta_\kappa$. With $\theta_\kappa = \mathcal{O}(\kappa)$ and $m = \mathcal{O}(\kappa^2)$ so that $c = \mathcal{O}(\kappa^2)$, we find that $\lambda^s < 1$. In addition, since $\eta^s < 1/L$ is still needed to guarantee convergence (cf. Theorem 7.2), one must have $\theta_\kappa > \kappa$. □

7.8 More on Numerical Experiments

7.8.1 More Numerical Experiments of Section 7.2.4

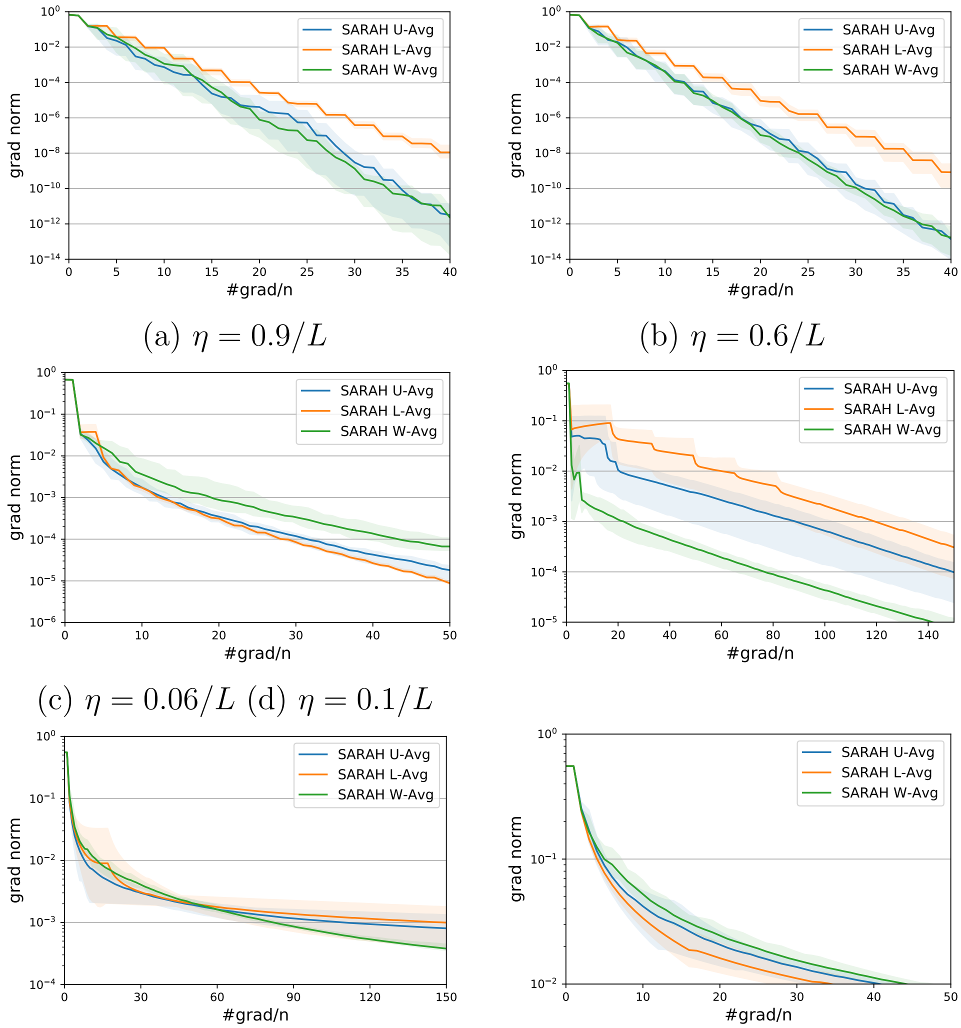
This section presents additional numerical experiments to support that averaging is not merely a “proof trick”. Specifically, experiments with SARAH under different types of averaging on datasets *a9a* and *diabetes* are showcased in Figure 7.6. Similar to the performance of SARAH on dataset *w7a*, W-Avg is better when the step size is chosen large, while a smaller step size favors L-Avg.

7.8.2 Details of Datasets Used in Section 7.4

The dimension d, number of training data n, the weight used for regularization, and other details of datasets used in Section 7.4 are listed in Table 7.1.

Table 7.1: Parameters of datasets used in numerical experiments

Dataset	d	n (train)	density	n (test)	μ
a9a	122	$3,185$	11.37%	$29,376$	0.001
rcv1	$47,236$	$20,242$	0.157%	$677,399$	0.00025
real-sim	$20,958$	$50,617$	0.24%	$21,692$	0.00025



(a) $\eta = 0.9/L$

(b) $\eta = 0.6/L$

(c) $\eta = 0.06/L$ (d) $\eta = 0.1/L$

(e) $\eta = 0.01/L$

(f) $\eta = 0.005/L$

Figure 7.6: Comparing SARAH with different types of averaging on datasets *a9a* and *diabetes*. In all experiments, we set $\mu = 0.002$ with $m = 5\kappa$.

CHAPTER 8

ENHANCING PARAMETER-FREE FRANK WOLFE WITH AN EXTRA SUBPROBLEM

8.1 Preliminaries

Notation. In Chapter 8, bold lowercase (uppercase) letters denote vectors (matrices); $\|\mathbf{x}\|$ stands for a norm of $\mathbf{x}$, with its dual norm written as $\|x\|_*$; and $\langle \mathbf{x}, \mathbf{y} \rangle$ denotes the inner product of vectors $\mathbf{x}$ and $\mathbf{y}$. We also define $x \wedge y := \min\{x, y\}$.

This section reviews FW and AFW in order to illustrate the proposed algorithm in a principled manner. We first pinpoint the class of problems to focus on.

Assumption 8.1. *(Lipschitz Continuous Gradient.) The function $f : \mathcal{X} \rightarrow \mathbb{R}$ has L-Lipchitz continuous gradients; that is, $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_* \leq L\|\mathbf{x} - \mathbf{y}\|, \forall\, \mathbf{x}, \mathbf{y} \in \mathcal{X}$.*

Assumption 8.2. *(Convex Objective Function.) The function $f : \mathcal{X} \rightarrow \mathbb{R}$ is convex; that is, $f(\mathbf{y}) - f(\mathbf{x}) \geq \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle, \forall\, \mathbf{x}, \mathbf{y} \in \mathcal{X}$.*

Assumption 8.3. *(Constraint Set.) The constraint set $\mathcal{X}$ is convex and compact with diameter D; that is, $\|\mathbf{x} - \mathbf{y}\| \leq D, \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$.*

Assumptions 8.1 – 8.3 are standard for FW type algorithms and will be taken to hold true throughout. A blackbox optimization paradigm is considered in this work, where the objective function and constraint set can be accessed through oracles only. In particular, the first-order oracle (FO) and the linear minimization oracle (LMO) are needed.

Definition 8.1. *(FO.) The first-order oracle takes $\mathbf{x} \in \mathcal{X}$ as an input and returns its gradient $\nabla f(\mathbf{x})$.*

Definition 8.2. *(LMO.) The linear minimization oracle takes a vector $\mathbf{g} \in \mathbb{R}^d$ as an input and returns a minimizer of $\min_{\mathbf{x} \in \mathcal{X}} \langle \mathbf{g}, \mathbf{x} \rangle$.*

Algorithm 8.1: FW [147]

```
Initialize: $\mathbf{x}_0 \in \mathcal{X}$
for $k = 0, 1, \ldots, K-1$ do
    $\mathbf{v}_{k+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \langle \nabla f(\mathbf{x}_k), \mathbf{x} \rangle$
    $\mathbf{x}_{k+1} = (1-\delta_k)\mathbf{x}_k + \delta_k \mathbf{v}_{k+1}$
end for
Return: $\mathbf{x}_K$
```

Except for gradients, problem dependent parameters such as function value, smoothness constant L, and constraint diameter D are not provided by FO and LMO. Hence, algorithms relying only on FO and LMO are parameter-free. Next, we recap FW and AFW with parameter-free step sizes to gain more insights for the proposed algorithm.

FW recap. FW is summarized in Algorithm 8.1. A subproblem with a linear loss, referred to also as an *FW step*, is solved per iteration via LMO. The FW step can be explained as finding a minimizer over $\mathcal{X}$ for the following supporting hyperplane of $f(\mathbf{x})$,

$$f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle. \tag{8.1}$$

Note that (8.1) is also a lower bound for $f(\mathbf{x})$ due to convexity. Upon obtaining $\mathbf{v}_{k+1}$ by minimizing (8.1), over $\mathcal{X}$, $\mathbf{x}_{k+1}$ is updated as a convex combination of $\mathbf{v}_{k+1}$ and $\mathbf{x}_k$ to eliminate the projection. The parameter-free step size is usually chosen as $\delta_k = \frac{2}{k+2}$. As for convergence, FW guarantees $f(\mathbf{x}_k) - f(\mathbf{x}^*) = \mathcal{O}(\frac{LD^2}{k})$.

AFW recap. As an FW variant, AFW in Algorithm 8.2 relies on Nesterov momentum type update: that is, it uses an auxiliary variable $\mathbf{y}_k$ to estimate $\mathbf{x}_{k+1}$ and calculates the gradient $\nabla f(\mathbf{y}_k)$. If one writes $\mathbf{g}_{k+1}$ explicitly, $\mathbf{v}_{k+1}$ can be equivalently described as a minimizer over $\mathcal{X}$ of the hyperplane

$$\sum_{\tau=0}^{k} w_k^\tau \big[f(\mathbf{y}_\tau) + \langle \nabla f(\mathbf{y}_\tau), \mathbf{x} - \mathbf{y}_\tau \rangle \big], \tag{8.2}$$

where $w_k^\tau = \delta_\tau \prod_{j=\tau+1}^{k} (1-\delta_j)$ and $\sum_{\tau=0}^{k} w_k^\tau \approx 1$ (the sum depends on the choice of δ_0). Note that $f(\mathbf{y}_\tau) + \langle \nabla f(\mathbf{y}_\tau), \mathbf{x} - \mathbf{y}_\tau \rangle$ is a supporting hyperplane of $f(\mathbf{x})$ at $\mathbf{y}_\tau$, hence (8.2) is a lower bound for $f(\mathbf{x})$ constructed through a weighted average of supporting hyperplanes at $\{\mathbf{y}_\tau\}$. AFW converges at $\mathcal{O}\big(\frac{LD^2}{k}\big)$ on general problems. When the constraint set is an active ℓ_2 norm

Algorithm 8.2: AFW [163]

```
Initialize: $\mathbf{x}_0 \in \mathcal{X}$, $\mathbf{g}_0 = \mathbf{0}$
for $k = 0, 1, \ldots, K-1$ do
    $\mathbf{y}_k = (1-\delta_k)\mathbf{x}_k + \delta_k \mathbf{v}_k$
    $\mathbf{g}_{k+1} = (1-\delta_k)\mathbf{g}_k + \delta_k \nabla f(\mathbf{y}_k)$
    $\mathbf{v}_{k+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \langle \mathbf{g}_{k+1}, \mathbf{x} \rangle$
    $\mathbf{x}_{k+1} = (1-\delta_k)\mathbf{x}_k + \delta_k \mathbf{v}_{k+1}$
end for
Return: $\mathbf{x}_K$
```

Algorithm 8.3: ExtraFW

```
1: Initialize: $\mathbf{x}_0$, $\mathbf{g}_0 = \mathbf{0}$, and $\mathbf{v}_0 = \mathbf{x}_0$
2: for $k = 0, 1, \ldots, K-1$ do
3:     $\mathbf{y}_k = (1-\delta_k)\mathbf{x}_k + \delta_k \mathbf{v}_k$ {prediction}
4:     $\hat{\mathbf{g}}_{k+1} = (1-\delta_k)\mathbf{g}_k + \delta_k \nabla f(\mathbf{y}_k)$
5:     $\hat{\mathbf{v}}_{k+1} = \arg\min_{\mathbf{v}\in\mathcal{X}} \langle \hat{\mathbf{g}}_{k+1}, \mathbf{v} \rangle$
6:     $\mathbf{x}_{k+1} = (1-\delta_k)\mathbf{x}_k + \delta_k \hat{\mathbf{v}}_{k+1}$ {correction}
7:     $\mathbf{g}_{k+1} = (1-\delta_k)\mathbf{g}_k + \delta_k \nabla f(\mathbf{x}_{k+1})$
8:     $\mathbf{v}_{k+1} = \arg\min_{\mathbf{v}\in\mathcal{X}} \langle \mathbf{g}_{k+1}, \mathbf{v} \rangle$ {extra FW step}
9: end for
10: Return: $\mathbf{x}_K$
```

ball, AFW has a faster rate $\mathcal{O}\big(\frac{LD^2}{k} \wedge \frac{TLD^2 \ln k}{k^2}\big)$, where T depends on D. Writing this rate compactly as $\mathcal{O}\big(\frac{TLD^2 \ln k}{k^2}\big)$, it is observed that AFW achieves acceleration with the price of a worse dependence on other parameters hidden in T. However, even for the k-dependence, AFW is $\mathcal{O}(\ln k)$ times slower compared with other momentum based algorithms such as NAG. This slowdown is because that the lower bound (8.2) is constructed based on $\{\mathbf{y}_k\}$ which are estimated $\{\mathbf{x}_{k+1}\}$. We will show that relying on a lower bound constructed using $\{\mathbf{x}_{k+1}\}$ directly, it is possible to avoid this $\mathcal{O}(\ln k)$ slowdown.

8.2 ExtraFW

This section introduces the main algorithm, ExtraFW, and establishes its constraint dependent faster rates.

8.2.1 Algorithm Design

ExtraFW is summarized in Algorithm 8.3. Different from the vanilla FW and AFW, two FW steps (lines 5 and 8 of Algorithm 8.3) are required per iteration. Compared with other algorithms relying on two gradient evaluations, such as Mirror-Prox [166, 167], ExtraFW reduces the computational burden of the projection. In addition, as an FW variant, ExtraFW can capture the properties such as sparsity or low rank promoted by the constraints more effectively through the update than those projection based algorithms. Detailed elaboration can be found in Section 8.3 and Section 8.7. To facilitate comparison with FW and AFW, ExtraFW is explained through constructing lower bounds of $f(\mathbf{x})$ in a "prediction-correction" manner. The merits of the PC update compared with AFW are: i) the elimination of $\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$ in analysis; and ii) it improves the convergence rate on certain class of problems as we will see later.

Lower bound prediction. Similar to AFW, the auxiliary variable $\mathbf{y}_k$ in line 3 of Algorithm 8.3 can be viewed as an estimate of $\mathbf{x}_{k+1}$. The first gradient is evaluated at $\mathbf{y}_k$ and is incorporated into $\hat{\mathbf{g}}_{k+1}$ which is an estimate of the weighted average of $\{\nabla f(\mathbf{x})_\tau\}_{\tau=1}^{k+1}$. By expanding $\hat{\mathbf{g}}_{k+1}$, one can verify that $\hat{\mathbf{v}}_{k+1}$ can be obtained equivalently through minimizing the following weighted sum,

$$\sum_{\tau=0}^{k-1} w_k^\tau \Big[f(\mathbf{x}_{\tau+1}) + \langle \nabla f(\mathbf{x}_{\tau+1}), \mathbf{x} - \mathbf{x}_{\tau+1}\rangle\Big] + \delta_k \Big[f(\mathbf{y}_k) + \big\langle \nabla f(\mathbf{y}_k), \mathbf{x} - \mathbf{y}_k\big\rangle\Big], \tag{8.3}$$

where $w_\tau = \delta_\tau \prod_{j=\tau+1}^{k}(1-\delta_j)$ and $\sum_{\tau=0}^{k-1} w_\tau + \delta_k \approx 1$. Note that each term inside square brackets forms a supporting hyperplane of $f(\mathbf{x})$, hence (8.3) is an (approximated) lower bound of $f(\mathbf{x})$ because of convexity. As a prediction to $f(\mathbf{x}_{k+1}) + \langle \nabla f(\mathbf{x}_{k+1}), \mathbf{x} - \mathbf{x}_{k+1}\rangle$, the last bracket in (8.3) will be corrected once $\mathbf{x}_{k+1}$ is obtained.

Lower bound correction. The gradient $\nabla f(\mathbf{x}_{k+1})$ is used to obtain a weighted averaged gradients $\mathbf{g}_{k+1}$. By unrolling $\mathbf{g}_{k+1}$, one can find that $\mathbf{v}_{k+1}$ is a minimizer of the following (approximated) lower bound of $f(\mathbf{x})$

$$\begin{aligned}\sum_{\tau=0}^{k-1} w_k^\tau \Big[f(\mathbf{x}_{\tau+1}) + \big\langle \nabla f(\mathbf{x}_{\tau+1}), \mathbf{x} - \mathbf{x}_{\tau+1}\big\rangle\Big] \\ + \delta_k \Big[f(\mathbf{x}_{k+1}) + \big\langle \nabla f(\mathbf{x}_{k+1}), \mathbf{x} - \mathbf{x}_{k+1}\big\rangle\Big].\end{aligned} \tag{8.4}$$

Comparing (8.3) and (8.4), we deduce that the terms in the last bracket of

(8.3) are corrected to the true supporting hyperplane of $f(\mathbf{x})$ at $\mathbf{x}_{k+1}$. In sum, the FW steps in ExtraFW rely on lower bounds of $f(\mathbf{x})$ constructed in a weighted average manner similar to AFW. However, the key difference is that ExtraFW leverages the supporting hyperplanes at true variables $\{\mathbf{x}_k\}$ rather than the auxiliary ones $\{\mathbf{y}_k\}$ in AFW through a "correction" effected by (8.4). In the following sections, we will show that the PC update in ExtraFW performs no worse than FW or AFW on general problems, while harnessing its own analytical merits on certain constraint sets.

8.2.2 Convergence of ExtraFW

We investigate the convergence of ExtraFW by considering the general case first. The analysis relies on the notion of ES introduced in [125]. An ES "estimates" f using a sequence of surrogate functions $\{\Phi_k(\mathbf{x})\}$ that are analytically tractable (e.g., being quadratic or linear). ES is formalized in the following definition.

Definition 8.3. *A tuple* $\big(\{\Phi_k(\mathbf{x})\}_{k=0}^{\infty}, \{\lambda_k\}_{k=0}^{\infty}\big)$ *is called an estimate sequence of function* $f(\mathbf{x})$ *if* $\lim_{k\to\infty}\lambda_k = 0$ *and for any* $\mathbf{x} \in \mathcal{X}$ *we have* $\Phi_k(\mathbf{x}) \leq (1-\lambda_k)f(\mathbf{x}) + \lambda_k\Phi_0(\mathbf{x})$.

The construction of ES varies for different algorithms (see e.g., [236, 125, 240, 10]). However, the reason to rely on the ES based analysis is similar, as summarized in the following lemma.

Lemma 8.1. *For* $\big(\{\Phi_k(\mathbf{x})\}_{k=0}^{\infty}, \{\lambda_k\}_{k=0}^{\infty}\big)$ *satisfying the definition of ES, if* $f(\mathbf{x}_k) \leq \min_{\mathbf{x}\in\mathcal{X}} \Phi_k(\mathbf{x}) + \xi_k, \forall k$, *it is true that*

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \lambda_k\big(\Phi_0(\mathbf{x}^*) - f(\mathbf{x}^*)\big) + \xi_k, \forall\, k.$$

As shown in Lemma 8.1, λ_k and ξ_k jointly characterize the convergence rate of $f(\mathbf{x}_k)$. Consider $\lambda_k = \mathcal{O}(\frac{1}{k})$ and $\xi_k = \mathcal{O}(\frac{1}{k})$ for an example. Keeping Lemma 8.1 in mind, we construct *two* sequences of *linear* surrogate functions for analyzing ExtraFW, which highlight the differences of our analysis with existing ES based approaches

$$\Phi_0(\mathbf{x}) = \hat{\Phi}_0(\mathbf{x}) \equiv f(\mathbf{x}_0), \tag{8.5a}$$

$$\hat{\Phi}_{k+1}(\mathbf{x}) = (1-\delta_k)\Phi_k(\mathbf{x}) + \delta_k\big[f(\mathbf{y}_k) + \langle \nabla f(\mathbf{y}_k), \mathbf{x} - \mathbf{y}_k\rangle\big],\ \forall k \geq 0, \tag{8.5b}$$

$$\Phi_{k+1}(\mathbf{x}) = (1-\delta_k)\Phi_k(\mathbf{x}) + \langle \nabla f(\mathbf{x}_{k+1}), \mathbf{x} - \mathbf{x}_{k+1}\rangle\big],\ \forall k \geq 0. \tag{8.5c}$$

Clearly, both $\Phi_k(\mathbf{x})$ and $\hat{\Phi}_k(\mathbf{x})$ are linear in $\mathbf{x}$, in contrast to the quadratic surrogate functions adopted for analyzing NAG [125]. Such linear surrogate functions are constructed specifically for FW type algorithms taking advantage of the compact and convex constraint set. Next we show that (8.5) and proper $\{\lambda_k\}$ form two different ES of f.

Lemma 8.2. *If we choose $\lambda_0 = 1$, $\delta_k \in (0,1)$, and $\lambda_{k+1} = (1-\delta_k)\lambda_k\ \forall k \geq 0$, both $\big(\{\Phi_k(\mathbf{x})\}_{k=0}^{\infty}, \{\lambda_k\}_{k=0}^{\infty}\big)$ and $\big(\{\hat{\Phi}_k(\mathbf{x})\}_{k=0}^{\infty}, \{\lambda_k\}_{k=0}^{\infty}\big)$ satisfy the definition of ES.*

The key reason behind the construction of surrogate functions in (8.5) is that they are closedly linked with the lower bounds (8.3) and (8.4) used in the FW steps, as stated in the next lemma.

Lemma 8.3. *Let $\mathbf{g}_0 = \mathbf{0}$, then it is true that $\mathbf{v}_k = \arg\min_{\mathbf{x}\in\mathcal{X}} \Phi_k(\mathbf{x})$ and $\hat{\mathbf{v}}_k = \arg\min_{\mathbf{x}\in\mathcal{X}} \hat{\Phi}_k(\mathbf{x})$.*

After relating the surrogate functions in (8.5) with ExtraFW, exploiting the analytical merits of the surrogate functions $\Phi_k(\mathbf{x})$ and $\hat{\Phi}_k(\mathbf{x})$, including being linear, next we show that $f(\mathbf{x}_k) \leq \min_{\mathbf{x}\in\mathcal{X}} \Phi_k(\mathbf{x}) + \xi_k, \forall k$, which is the premise of Lemma 8.1.

Lemma 8.4. *Let $\xi_0 = 0$ and other parameters chosen the same as previous lemmas. Denote $\Phi_k^* := \Phi_k(\mathbf{v}_k)$ as the minimum value of $\Phi_k(\mathbf{x})$ over $\mathcal{X}$ (cf. Lemma 8.3), then ExtraFW guarantees that for any $k \geq 0$*

$$f(\mathbf{x}_k) \leq \Phi_k^* + \xi_k,\ \textit{with}\ \xi_{k+1} = (1-\delta_k)\xi_k + \frac{3LD^2}{2}\delta_k^2.$$

Based on Lemma 8.4, the value of $f(\mathbf{x}_k)$ and Φ_k^* can be used to derive the stopping criterion if one does not want to preset the iteration number K. Further discussions are provided in Section 8.4.6. Now we are ready to apply Lemma 8.1 to establish the convergence of ExtraFW.

Theorem 8.1. *Suppose that Assumptions 8.1, 8.2, and 8.3 are satisfied. Choosing $\delta_k = \frac{2}{k+3}$, and $\mathbf{g}_0 = \mathbf{0}$, ExtraFW in Algorithm 8.3 guarantees*

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) = \mathcal{O}\bigg(\frac{LD^2}{k}\bigg), \forall k.$$

This convergence rate of ExtraFW has the same order as AFW and FW. In addition, Theorem 8.1 translates into $\mathcal{O}(\frac{LD^2}{\epsilon})$ queries of LMO to ensure $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \epsilon$, which matches to the lower bound [155, 148].

The obstacle for faster rates. As shown in the detailed proof, one needs to guarantee that either $\|\mathbf{v}_k - \hat{\mathbf{v}}_{k+1}\|^2$ or $\|\mathbf{v}_{k+1} - \hat{\mathbf{v}}_{k+1}\|^2$ is small enough to obtain a faster rate than Theorem 8.1. This is difficult in general because there could be multiple $\mathbf{v}_k$ and $\hat{\mathbf{v}}_k$ solving the FW steps. A simple example is to consider the ith entry $[\mathbf{g}_k]_i = 0$. The ith entry $[\mathbf{v}_k]_i$ can then be chosen arbitrarily as long as $\mathbf{v}_k \in \mathcal{X}$. The non-uniqueness of $\mathbf{v}_k$ prevents one from ensuring a small upper bound of $\|\mathbf{v}_k - \hat{\mathbf{v}}_{k+1}\|^2$, $\forall\ \mathbf{v}_k$. In spite of this, we will show that together with the structure on $\mathcal{X}$, ExtraFW can attain faster rates.

8.2.3 Acceleration of ExtraFW

In this section, we provide constraint-dependent accelerated rates of ExtraFW when $\mathcal{X}$ is some norm ball. Even for projection based algorithms, most of faster rates are obtained with step sizes depending on L [165, 166]. Thus, faster rates for parameter-free algorithms are challenging to establish. An extra assumption is needed in this section.

Assumption 8.4. *The constraint is active, i.e., $\|\nabla f(\mathbf{x}^*)\|_2 \geq G > 0$.*

It is natural to rely on the position of the optimal solution in FW type algorithms for analysis, and one can see this assumption also in [160, 242, 163, 243]. For a number of machine learning tasks, Assumption 8.4 is rather mild. Relying on Lagrangian duality, it can be seen that problem (1.4) with a norm ball constraint is equivalent to the regularized formulation $\min_{\mathbf{x}} f(\mathbf{x}) + \gamma g(\mathbf{x})$, where $\gamma \geq 0$ is the Lagrange multiplier, and $g(\mathbf{x})$ denotes some norm. In view of this, Assumption 8.4 simply implies that $\gamma > 0$ in the equivalent regularized formulation, that is, the norm ball constraint plays the role of a regularizer. Given the prevalence of the regularized formulation in machine learning, it is worth investigating its equivalent constrained form (1.4) under Assumption 8.4.

Technically, the need behind Assumption 8.4 can be exemplified through a one-dimensional problem. Consider minimizing $f(x) = x^2$ over $\mathcal{X} = \{x | x \in [-1, 1]\}$. We clearly have $x^* = 0$ for which the constraint is inactive at the optimal solution. Recall a faster rate of ExtraFW requires $\|\hat{v}_{k+1} - v_{k+1}\|_2$ to be small. When x_k is close to $x^* = 0$, it can happen that $\hat{g}_{k+1} > 0$ and $g_{k+1} < 0$, leading to $\hat{v}_{k+1} = -1$ and $v_{k+1} = 1$. The faster rate is prevented by pushing v_{k+1} and $\hat{v}_{k+1}$ further apart from each other.

Next, we consider different instances of norm ball constraints as examples to the acceleration of ExtraFW. For simplicity of exposition, the intuition and technical details are discussed using an ℓ_2 norm ball constraint in the main experiment. Detailed analysis for ℓ_1 and n-support norm ball [244] constraints are provided in Section 8.6.1 and 8.6.2.

ℓ_2 **norm ball constraint.** Consider $\mathcal{X} := \{\mathbf{x} | \|\mathbf{x}\|_2 \leq \frac{D}{2}\}$. In this case, $\mathbf{v}_{k+1}$ and $\hat{\mathbf{v}}_{k+1}$ admit closed-form solutions, taking $\mathbf{v}_{k+1}$ as an example,

$$\mathbf{v}_{k+1} = \arg\min_{\mathbf{x} \in \mathcal{X}} \langle \mathbf{g}_{k+1}, \mathbf{x} \rangle = -\frac{D}{2\|\mathbf{g}_{k+1}\|_2} \mathbf{g}_{k+1}. \tag{8.6}$$

We assume that when using $\mathbf{g}_{k+1}$ as the input to the LMO, the returned vector is given by (8.6). This is reasonable since it is what we usually implemented in practice. Though it rarely happens, one can choose $\mathbf{v}_{k+1} = \hat{\mathbf{v}}_{k+1}$ to proceed if $\mathbf{g}_{k+1} = \mathbf{0}$. Similarly, we can simply set $\hat{\mathbf{v}}_{k+1} = \mathbf{v}_k$ if $\hat{\mathbf{g}}_{k+1} = \mathbf{0}$. The uniqueness of $\mathbf{v}_{k+1}$ is ensured by its closed-form solution, wiping out the obstacle for a faster rate.

Theorem 8.2. *Suppose that Assumptions 8.1, 8.2, 8.3, and 8.4 are satisfied, and $\mathcal{X}$ is an ℓ_2 norm ball. Choosing $\delta_k = \frac{2}{k+3}$ and $\mathbf{g}_0 = \mathbf{0}$, ExtraFW in Algorithm 8.3 guarantees*

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) = \mathcal{O}\bigg(\frac{LD^2}{k} \wedge \frac{LD^2 T}{k^2}\bigg), \forall k,$$

where T is a constant depending only on L, G, and D.

Theorem 8.2 admits a couple of interpretations. By writing the rate compactly, ExtraFW achieves accelerated rate $\mathcal{O}\big(\frac{TLD^2}{k^2}\big), \forall k$ with a worse dependence on D compared to the vanilla FW. Or alternatively, the "asymptotic" performance at $k \geq T$ is strictly improved over the vanilla FW. It is worth mentioning that the choices of δ_k and $\mathbf{g}_0$ are not changed compared to Theorem 8.1 so that the parameter-free implementation is the same regardless whether accelerated. In other words, prior knowledge on whether Assumption 8.4 holds is not needed in practice. Compared with CGS, ExtraFW sacrifices the D dependence in the convergence rate to trade for i) the non-necessity of the knowledge of L and D and ii) ensuring two FW subproblems per iteration (whereas at most $\mathcal{O}(k)$ subproblems are needed in CGS). When comparing with AFW [163], the convergence rate of ExtraFW is improved by a factor of $\mathcal{O}(\ln k)$, and the analysis does not rely on the constant $M := \max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$.

ℓ_1 **norm ball constraint.** For the sparsity-promoting constraint $\mathcal{X} := \{\mathbf{x} | \|\mathbf{x}\|_1 \leq R\}$, the FW steps can be solved in closed form too. Taking $\mathbf{v}_{k+1}$ as an example, we have

$$\mathbf{v}_{k+1} = R \cdot [0, \ldots, 0, -\text{sgn}[\mathbf{g}_{k+1}]_i, 0, \ldots, 0]^\top \text{ with } i = \arg\max_j |[\mathbf{g}_{k+1}]_j|. \tag{8.7}$$

We show in Theorem 8.3 (see Section 8.6.1) that when Assumption 8.4 holds and the set $\arg\max_j \left|[\nabla f(\mathbf{x}^*)]_j\right|$ has cardinality 1, a faster rate $\mathcal{O}(\frac{T_1 L D^2}{k^2})$ can be obtained with the constant T_1 depending on L, G, and D. The additional assumption here is known as *strict complementarity* and has been adopted also in, e.g.,[245, 246].

n**-support norm ball constraint.** The n-support norm ball is a tighter relaxation of a sparsity prompting ℓ_0 norm ball combined with an ℓ_2 norm penalty compared with the ElasticNet [247]. It is defined as $\mathcal{X} := \text{conv}\{\mathbf{x} | \|\mathbf{x}\|_0 \leq n, \|\mathbf{x}\|_2 \leq R\}$, where $\text{conv}\{\cdot\}$ denotes the convex hull [244]. The closed-form solution of $\mathbf{v}_{k+1}$ is given by [248]

$$\mathbf{v}_{k+1} = -\frac{R}{\|\text{top}_n(\mathbf{g}_{k+1})\|_2} \text{top}_n(\mathbf{g}_{k+1}), \tag{8.8}$$

where $\text{top}_n(\mathbf{g})$ denotes the truncated version of $\mathbf{g}$ with its top n (in magnitude) entries preserved. A faster rate $\mathcal{O}(\frac{T_2 L D^2}{k^2})$ is guaranteed by ExtraFW under Assumption 8.4 and a condition similar to strict complementarity (see Theorem 8.4 in the Section 8.6.2). Again, the constant T_2 here depends on L, G, and D.

Other constraints. Note that the faster rates for ExtraFW are not limited to the exemplified constraint sets. In principle, if i) certain structure such as sparsity is promoted by the constraint set so that $\mathbf{x}^*$ is likely to lie on the boundary of $\mathcal{X}$, and ii) one can ensure the uniqueness of $\mathbf{v}_k$ through either a closed-form solution or a specific implementation manner, the acceleration of ExtraFW is achievable. Discussions for faster rates on a simplex $\mathcal{X}$ can be found in Section 8.6.1. In addition, one can easily extend our results to the matrix case where the constraint set is the Frobenius or the nuclear norm ball since they are ℓ_2 and ℓ_1 norms on the singular values of matrices, respectively.

8.3 Numerical Experiments

This section deals with numerical experiments of ExtraFW to showcase its effectiveness on different machine learning problems. Due to the space limitation, details of the datasets and implementation are deferred to Section 8.7. For comparison, the benchmarked algorithms are chosen as: i) GD with standard step size $\frac{1}{L}$; ii) NAG with step sizes in [144]; iii) FW with parameter-free step size $\frac{2}{k+2}$ [148]; and iv) AFW with step size $\frac{2}{k+3}$ [163].

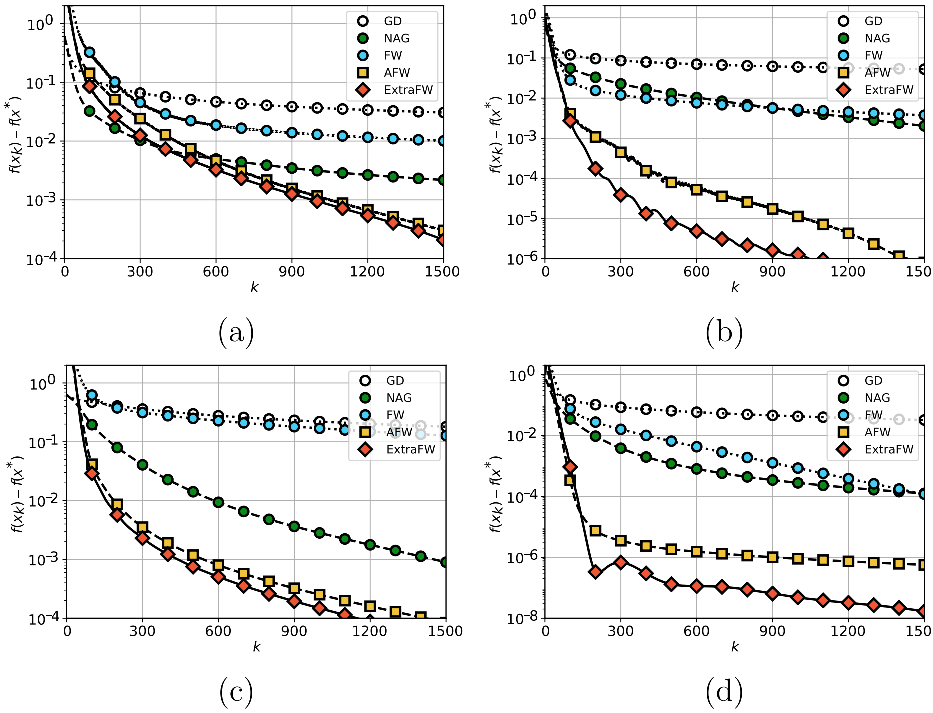
8.3.1 Binary Classification

We first investigate the performance of ExtraFW on binary classification using logistic regression. The constraints considered include: i) ℓ_2 norm ball for generalization merits and ii) ℓ_1 and n-support norm ball for promoting a sparse solution. The objective function is

$$f(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \ln\big(1 + \exp(-b_i\langle \mathbf{a}_i, \mathbf{x}\rangle)\big), \tag{8.9}$$

where $(\mathbf{a}_i, b_i)$ is the (feature, label) pair of datum i, and N is the number of data. Datasets *mnist* and those from LIBSVM[1] are used in the numerical experiments. Figures reporting experiment accuracy and additional experiments are postponed into Section 8.7.

[1] `http://yann.lecun.com/exdb/mnist/`, and `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html`.



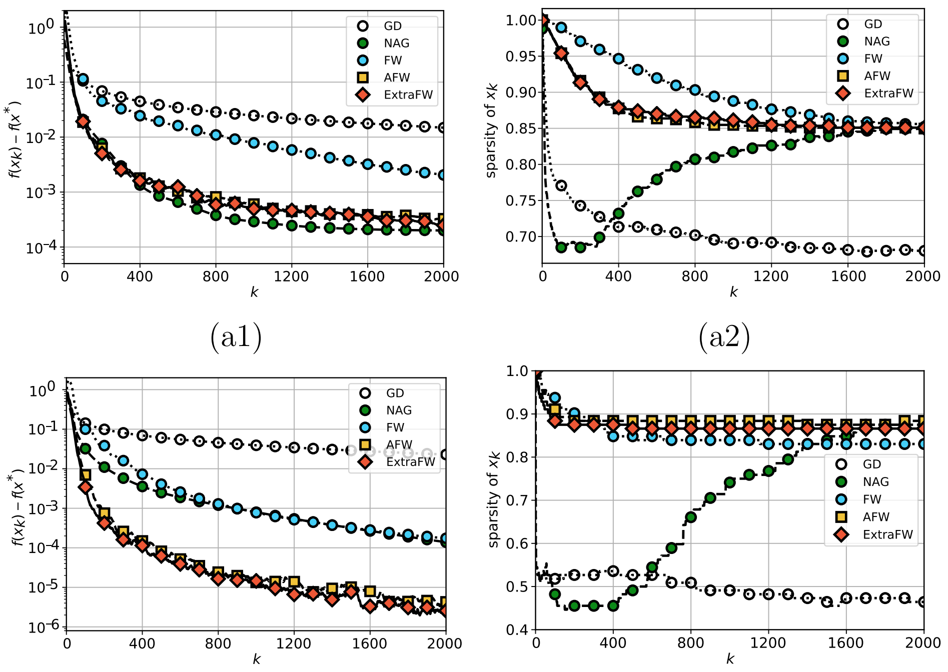
(a) (b)

(c) (d)

Figure 8.1: Performance of ExtraFW for binary classification with an ℓ_2 norm ball constraint on datasets: (a) *mnist*, (b) *w7a*, (c) *realsim*, and (d) *mushroom*.

ℓ_2 **norm ball constraint.** We start with $\mathcal{X} = \{\mathbf{x} | \|\mathbf{x}\|_2 \leq R\}$. The optimality errors are plotted in Fig. 8.1. On all tested datasets, ExtraFW outperforms AFW, NAG, FW, and GD, demonstrating the $\mathcal{O}(\frac{1}{k^2})$ convergence rate established in Theorem 8.2. In addition, the simulation also suggests that T is generally small for logistic loss. On dataset *w7a* and *mushroom*, ExtraFW is significantly faster than AFW. All these observations jointly confirm the usefulness of the extra gradient and the PC update.

ℓ_1 **norm ball constraint.** Let $\mathcal{X} = \{\mathbf{x} | \|\mathbf{x}\|_1 \leq R\}$ be the constraint set to promote sparsity on the solution. Note that FW type updates directly guarantee that $\mathbf{x}_k$ has at most k non-zero entries when initialized at $\mathbf{x}_0 = \mathbf{0}$; see detailed discussions in Section 8.7.2. In the simulation, R is tuned to obtain a solution that is almost as sparse as the dataset itself. The numerical results on datasets *mnist* and *mushroom* including both optimality error and the sparsity level of the solution can be found in Fig. 8.2. On dataset *mnist*, ExtraFW slightly outperforms AFW but is not as fast as NAG. However, ExtraFW consistently finds solutions sparser than NAG. While on dataset *mushroom*, it can be seen that both AFW and ExtraFW outperform NAG,



(a1)

(a2)

(b1)

(b2)

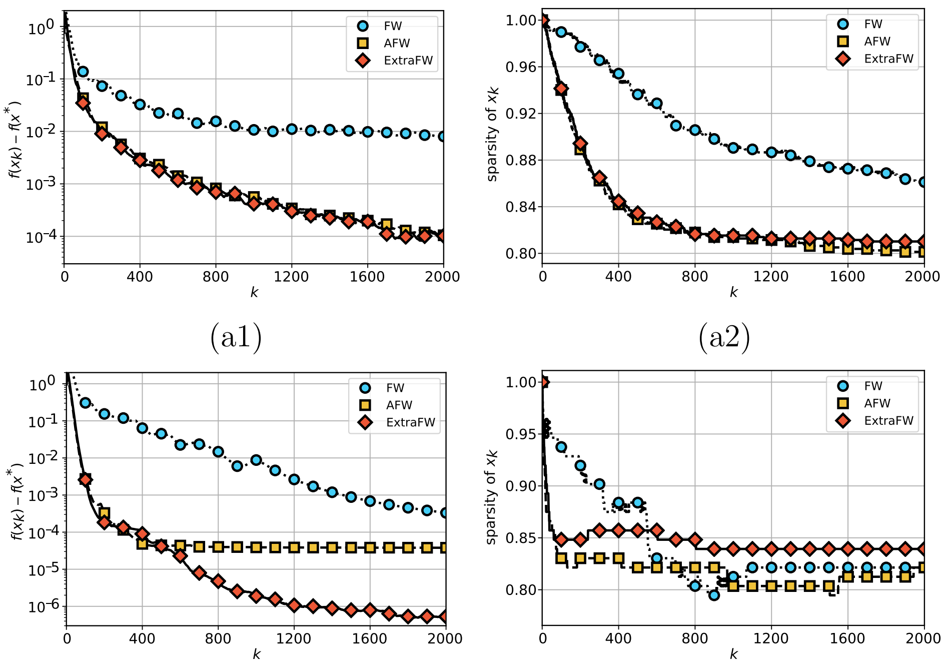
Figure 8.2: Performance of ExtraFW for binary classification with an ℓ_1 norm ball constraint: (a1) optimality error on *mnist*, (a2) solution sparsity on *mnist*, (b1) optimality error on *mushroom*, and, (b2) solution sparsity on *mushroom*.

with ExtraFW slightly faster than AFW. ExtraFW finds sparser solutions than NAG.

***n*-support norm ball constraint.** Effective projection onto such a constraint is unknown yet, hence GD and NAG are not included in the experiment. The performance of ExtraFW can be found in Fig. 8.3. On dataset *mnist*, both AFW and ExtraFW converge much faster than FW with ExtraFW slightly faster than AFW. However, FW trades the solution accuracy with its sparsity. On dataset *mushroom*, ExtraFW converges much faster than AFW and FW, while finding the sparsest solution.

8.3.2 Matrix Completion

We then consider matrix completion problems that are ubiquitous in recommender systems. Consider a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ with partially observed entries, that is, entries A_{ij} for $(i, j) \in \mathcal{K}$ are known, where $\mathcal{K} \subset \{1, \ldots, m\} \times$



(a1) (a2)

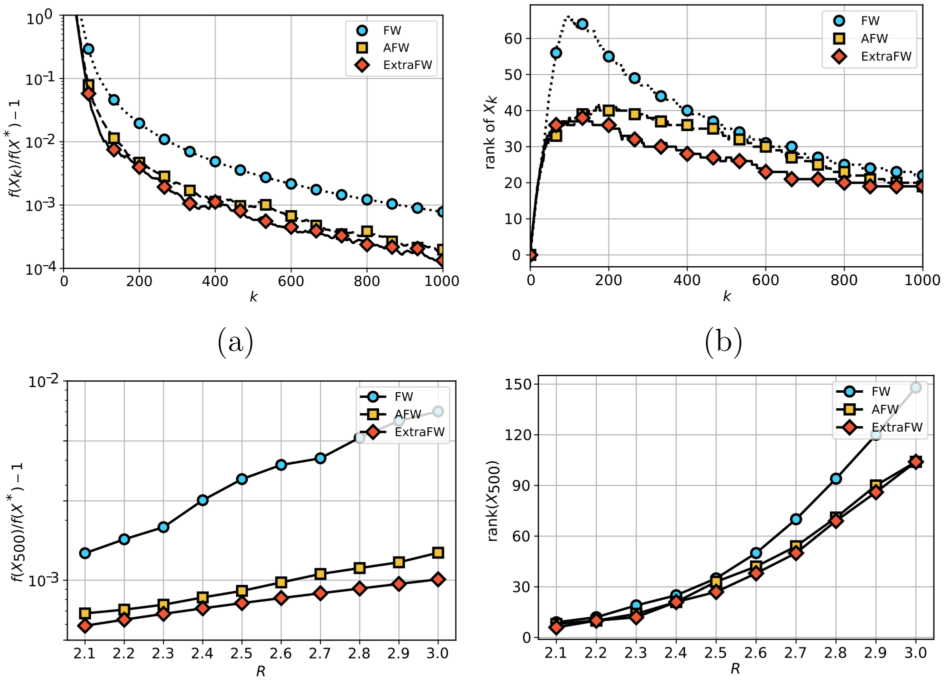
(b1) (b2)

Figure 8.3: Performance of ExtraFW for binary classification with an n-support norm ball constraint: (a1) optimality error on *mnist*, (a2) solution sparsity on *mnist*, (b1) optimality error on *mushroom*, and (b2) solution sparsity on *mushroom*.

$\{1, \ldots, n\}$. Note that the observed entries can also be contaminated by noise. The task is to predict the unobserved entries of $\mathbf{A}$. Although this problem can be approached in several ways, within the scope of recommender systems, a commonly adopted empirical observation is that $\mathbf{A}$ is low rank [249, 250, 251]. The objective boils down to

$$\min_{\mathbf{X}} \quad \frac{1}{2} \sum_{(i,j)\in\mathcal{K}} (X_{ij} - A_{ij})^2 \quad \text{s.t.} \quad \|\mathbf{X}\|_{\text{nuc}} \leq R, \tag{8.10}$$

where $\|\cdot\|_{\text{nuc}}$ denotes the nuclear norm. Problem (8.10) is difficult to be solved via GD or NAG because projection onto a nuclear norm ball requires to perform SVD, which has complexity $\mathcal{O}\big(mn(m \wedge n)\big)$. On the contrary, FW and its variants are more suitable for (8.10) given the facts: i) Assumptions 8.1 – 8.3 are satisfied under nuclear norm [142]; ii) FW step can be solved easily with complexity at the same order as the number of nonzero entries; and iii) the update promotes low-rank solution directly [142]. More on ii)



(a)

(b)

(c)

(d)

Figure 8.4: Performance of ExtraFW for matrix completion: (a) optimality vs k, (b) solution rank vs k, (c) optimality at $k = 500$ vs R, and, (d) solution rank at $k = 500$ vs R.

and iii) are discussed in Section 8.7.3.

We test ExtraFW on a widely used dataset, *MovieLens100K*[2]. The experiments follow the same steps in [142, Freund et al., 2017]. The numerical performance of ExtraFW, AFW, and FW can be found in Fig. 8.4. In Figures 8.4(a) and 8.4(b), we plot the optimality error and rank versus k choosing $R = 2.5$. It is observed that ExtraFW exhibits the best performance in terms of both optimality error and solution rank. In particular, ExtraFW roughly achieves 2.5x performance improvement compared with FW in terms of optimality error. We further compare the convergence of ExtraFW to AFW and FW at iteration $k = 500$ under different choices of R in Figures 8.4(c) and 8.4(d). It can be seen that ExtraFW still finds solutions with the lowest optimality error and rank. Moreover, the performance gap between ExtraFW and AFW increases with R, suggesting the inclined tendency of preferring ExtraFW over AFW and FW as R grows.

[2] `https://grouplens.org/datasets/movielens/100k/`

8.4 Proofs in Section 8.2.2

8.4.1 Proof of Lemma 8.1

Proof. If $f(\mathbf{x}_k) \leq \min_{\mathbf{x}\in\mathcal{X}} \Phi_k(\mathbf{x}) + \xi_k$ holds, then we have

$$f(\mathbf{x}_k) \leq \min_{\mathbf{x}\in\mathcal{X}} \Phi_k(\mathbf{x}) + \xi_k \leq \Phi_k(\mathbf{x}^*) + \xi_k \leq (1-\lambda_k)f(\mathbf{x}^*) + \lambda_k\Phi_0(\mathbf{x}^*) + \xi_k,$$

where the last inequality is because Definition 8.3. Subtracting $f(\mathbf{x}^*)$ on both sides, we arrive at

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \lambda_k\big(\Phi_0(\mathbf{x}^*) - f(\mathbf{x}^*)\big) + \xi_k,$$

which completes the proof. □

8.4.2 Proof of Lemma 8.2

Proof. We prove $\big(\{\Phi_k(\mathbf{x})\}_{k=0}^\infty,\{\lambda_k\}_{k=0}^\infty\big)$ is an ES of f by induction. Because $\lambda_0 = 1$, it holds that $\Phi_0(\mathbf{x}) = (1-\lambda_0)f(\mathbf{x}) + \lambda_0\Phi_0(\mathbf{x}) = \Phi_0(\mathbf{x})$. Suppose that $\Phi_k(\mathbf{x}) \leq (1-\lambda_k)f(\mathbf{x}) + \lambda_k\Phi_0(\mathbf{x})$ is true for some k. We have

$$\begin{aligned}\Phi_{k+1}(\mathbf{x}) &= (1-\delta_k)\Phi_k(\mathbf{x}) + \delta_k\Big[f(\mathbf{x}_{k+1}) + \big\langle\nabla f(\mathbf{x}_{k+1}), \mathbf{x}-\mathbf{x}_{k+1}\big\rangle\Big]\\ &\overset{(a)}{\leq} (1-\delta_k)\Phi_k(\mathbf{x}) + \delta_k f(\mathbf{x})\\ &\leq (1-\delta_k)\Big[(1-\lambda_k)f(\mathbf{x}) + \lambda_k\Phi_0(\mathbf{x})\Big] + \delta_k f(\mathbf{x})\\ &= (1-\lambda_{k+1})f(\mathbf{x}) + \lambda_{k+1}\Phi_0(\mathbf{x}),\end{aligned}$$

where (a) is because f is convex; and the last equation is by definition of λ_{k+1}. Together with the fact that $\lim_{k\to\infty}\lambda_k = 0$, one can see that the tuple $\big(\{\Phi_k(\mathbf{x})\}_{k=0}^\infty, \{\lambda_k\}_{k=0}^\infty\big)$ is an ES of f.

Next we show $\big(\{\hat{\Phi}_k(\mathbf{x})\}_{k=0}^\infty, \{\lambda_k\}_{k=0}^\infty\big)$ is also an ES. Clearly $\hat{\Phi}_0(\mathbf{x}) = (1-\lambda_0)f(\mathbf{x}) + \lambda_0\Phi_0(\mathbf{x}) = \hat{\Phi}_0(\mathbf{x})$. Next for $k \geq 0$, using similar arguments, we have

$$\begin{aligned}\hat{\Phi}_{k+1}(\mathbf{x}) &= (1-\delta_k)\Phi_k(\mathbf{x}) + \delta_k\Big[f(\mathbf{y}_k) + \big\langle\nabla f(\mathbf{y}_k), \mathbf{x}-\mathbf{y}_k\big\rangle\Big]\\ &\leq (1-\delta_k)\Phi_k(\mathbf{x}) + \delta_k f(\mathbf{x})\end{aligned}$$

$$
\begin{aligned}
&\leq (1-\delta_k)\Big[(1-\lambda_k)f(\mathbf{x}) + \lambda_k \Phi_0(\mathbf{x})\Big] + \delta_k f(\mathbf{x}) \\
&= (1-\lambda_{k+1})f(\mathbf{x}) + \lambda_{k+1}\Phi_0(\mathbf{x}) \\
&= (1-\lambda_{k+1})f(\mathbf{x}) + \lambda_{k+1}\hat{\Phi}_0(\mathbf{x}).
\end{aligned}
$$

The proof is thus completed. □

8.4.3 Proof of Lemma 8.3

Proof. For convenience, denote $B_k(\mathbf{x}) := f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k\rangle$. We can unroll $\Phi_{k+1}(\mathbf{x})$ as

$$
\begin{aligned}
\Phi_{k+1}(\mathbf{x}) &= (1-\delta_k)\Phi_k(\mathbf{x}) + \delta_k B_{k+1}(\mathbf{x}) \qquad (8.11) \\
&= (1-\delta_k)(1-\delta_{k-1})\Phi_{k-1}(\mathbf{x}) + (1-\delta_k)\delta_{k-1}B_k(\mathbf{x}) + \delta_k B_{k+1}(\mathbf{x}) \\
&= \Phi_0(\mathbf{x})\prod_{\tau=0}^{k}(1-\delta_\tau) + \sum_{\tau=0}^{k}\delta_\tau B_{\tau+1}(\mathbf{x})\prod_{j=\tau+1}^{k}(1-\delta_j) \\
&= f(\mathbf{x}_0)\prod_{\tau=0}^{k}(1-\delta_\tau) + \sum_{\tau=0}^{k}\delta_\tau B_{\tau+1}(\mathbf{x})\prod_{j=\tau+1}^{k}(1-\delta_j).
\end{aligned}
$$

Hence, the minimizer of $\Phi_{k+1}(\mathbf{x})$ can be rewritten as

$$
\begin{aligned}
&\arg\min_{\mathbf{x}\in\mathcal{X}}\ \Phi_{k+1}(\mathbf{x}) \\
&= \arg\min_{\mathbf{x}\in\mathcal{X}}\ f(\mathbf{x}_0)\prod_{\tau=0}^{k}(1-\delta_\tau) + \sum_{\tau=0}^{k}\delta_\tau B_{\tau+1}(\mathbf{x})\prod_{j=\tau+1}^{k}(1-\delta_j) \qquad (8.12) \\
&= \arg\min_{\mathbf{x}\in\mathcal{X}}\ \sum_{\tau=0}^{k}\delta_\tau\big[f(\mathbf{x}_{\tau+1}) + \langle\nabla f(\mathbf{x}_{\tau+1}), \mathbf{x}-\mathbf{x}_{\tau+1}\rangle\big] \times \prod_{j=\tau+1}^{k}(1-\delta_j) \\
&= \arg\min_{\mathbf{x}\in\mathcal{X}}\ \sum_{\tau=0}^{k}\delta_\tau\langle\nabla f(\mathbf{x}_{\tau+1}), \mathbf{x}\rangle\prod_{j=\tau+1}^{k}(1-\delta_j) \\
&= \arg\min_{\mathbf{x}\in\mathcal{X}}\ \sum_{\tau=0}^{k}\Big\langle\delta_\tau\nabla f(\mathbf{x}_{\tau+1})\prod_{j=\tau+1}^{k}(1-\delta_j), \mathbf{x}\Big\rangle \\
&= \arg\min_{\mathbf{x}\in\mathcal{X}}\ \langle\mathbf{g}_{k+1}, \mathbf{x}\rangle,
\end{aligned}
$$

where the last equation is because

$$\begin{aligned}
\mathbf{g}_{k+1} &= (1-\delta_k)\mathbf{g}_k + \delta_k \nabla f(\mathbf{x}_{k+1}) \\
&= (1-\delta_k)(1-\delta_{k-1})\mathbf{g}_{k-1} + (1-\delta_k)\delta_{k-1}\nabla f(\mathbf{x}_k) + \delta_k \nabla f(\mathbf{x}_{k+1}) \\
&= \mathbf{g}_0 \prod_{\tau=0}^{k}(1-\delta_\tau) + \sum_{\tau=0}^{k} \delta_\tau \nabla f(\mathbf{x}_{\tau+1}) \prod_{j=\tau+1}^{k}(1-\delta_j) \\
&= \sum_{\tau=0}^{k} \delta_\tau \nabla f(\mathbf{x}_{\tau+1}) \prod_{j=\tau+1}^{k}(1-\delta_j).
\end{aligned}$$

From (8.12) it is not hard to see $\mathbf{v}_{k+1}$ minimizes $\Phi_{k+1}(\mathbf{x})$.

If we write $\hat{\mathbf{g}}_{k+1}$ explicitly, we can obtain

$$\begin{aligned}
\hat{\Phi}_{k+1}(\mathbf{x}) &= (1-\delta_k)\Phi_k(\mathbf{x}) + \delta_k\Big[f(\mathbf{y}_k) + \big\langle \nabla f(\mathbf{y}_k), \mathbf{x}-\mathbf{y}_k \big\rangle\Big] \\
&= f(\mathbf{x}_0)\prod_{\tau=0}^{k}(1-\delta_\tau) + \sum_{\tau=0}^{k-1} \delta_\tau B_{\tau+1}(\mathbf{x}) \prod_{j=\tau+1}^{k}(1-\delta_j) \\
&\quad + \delta_k\Big[f(\mathbf{y}_k) + \big\langle \nabla f(\mathbf{y}_k), \mathbf{x}-\mathbf{y}_k \big\rangle\Big].
\end{aligned}$$

Hence using similar arguments as above we have

$$\begin{aligned}
\operatorname*{arg\,min}_{\mathbf{x}\in\mathcal{X}} \ \hat{\Phi}_{k+1}(\mathbf{x}) &= \operatorname*{arg\,min}_{\mathbf{x}\in\mathcal{X}} \ \Big\langle \delta_k \nabla f(\mathbf{y}_k) + \sum_{\tau=0}^{k-1} \delta_\tau \nabla f(\mathbf{x}_{\tau+1}) \prod_{j=\tau+1}^{k}(1-\delta_j), \mathbf{x} \Big\rangle \\
&= \operatorname*{arg\,min}_{\mathbf{x}\in\mathcal{X}} \ \langle \hat{\mathbf{g}}_{k+1}, \mathbf{x}\rangle = \hat{\mathbf{v}}_{k+1},
\end{aligned}$$

which implies that $\hat{\mathbf{v}}_{k+1}$ is a minimizer of $\hat{\Phi}_{k+1}(\mathbf{x})$ over $\mathcal{X}$. The lemma is thus proven. □

8.4.4 Proof of Lemma 8.4

Proof. We prove this lemma by induction. Since $\Phi_0(\mathbf{x}) \equiv f(\mathbf{x}_0)$ and $\xi_0 = 0$, it is clear that $f(\mathbf{x}_0) \le \Phi_0^* + \xi_0$.

Now suppose that $f(\mathbf{x}_k) \le \Phi_k^* + \xi_k$ holds for some $k > 0$, we will show

$f(\mathbf{x}_{k+1}) \leq \Phi_{k+1}^* + \xi_{k+1}$. To start with, we have from Assumption 7.1 that

$$\begin{aligned} f(\mathbf{x}_{k+1}) &\leq f(\mathbf{y}_k) + \big\langle \nabla f(\mathbf{y}_k), \mathbf{x}_{k+1} - \mathbf{y}_k \big\rangle + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{y}_k\|^2 \\ &\overset{(a)}{=} f(\mathbf{y}_k) + (1-\delta_k)\big\langle \nabla f(\mathbf{y}_k), \mathbf{x}_k - \mathbf{y}_k \big\rangle \\ &\qquad + \delta_k \big\langle \nabla f(\mathbf{y}_k), \hat{\mathbf{v}}_{k+1} - \mathbf{y}_k \big\rangle + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{y}_k\|^2 \\ &\overset{(b)}{=} f(\mathbf{y}_k) + (1-\delta_k)\big\langle \nabla f(\mathbf{y}_k), \mathbf{x}_k - \mathbf{y}_k \big\rangle \\ &\qquad + \delta_k \big\langle \nabla f(\mathbf{y}_k), \hat{\mathbf{v}}_{k+1} - \mathbf{y}_k \big\rangle + \frac{L\delta_k^2}{2}\|\hat{\mathbf{v}}_{k+1} - \mathbf{v}_k\|^2 \\ &\overset{(c)}{\leq} (1-\delta_k) f(\mathbf{x}_k) + \delta_k f(\mathbf{y}_k) + \delta_k \big\langle \nabla f(\mathbf{y}_k), \hat{\mathbf{v}}_{k+1} - \mathbf{y}_k \big\rangle \\ &\qquad + \frac{L\delta_k^2}{2}\|\hat{\mathbf{v}}_{k+1} - \mathbf{v}_k\|^2, \end{aligned} \tag{8.13}$$

where (a) is because $\mathbf{x}_{k+1} = (1-\delta_k)\mathbf{x}_k + \delta_k \hat{\mathbf{v}}_{k+1}$; (b) is by the choice of $\mathbf{x}_{k+1}$ and $\mathbf{y}_k$; and (c) is from convexity, that is, $\langle \nabla f(\mathbf{y}_k), \mathbf{x}_k - \mathbf{y}_k \rangle \leq f(\mathbf{x}_k) - f(\mathbf{y}_k)$. For convenience, we denote $\hat{\Phi}_k^* := \hat{\Phi}_k(\hat{\mathbf{v}}_k)$ as the minimum value of $\hat{\Phi}_k(\mathbf{x})$ over $\mathcal{X}$ (the equation here is the result of Lemma 8.3). Then we have

$$\begin{aligned} \hat{\Phi}_{k+1}^* = \hat{\Phi}_{k+1}(\hat{\mathbf{v}}_{k+1}) &\overset{(d)}{=} (1-\delta_k)\Phi_k(\hat{\mathbf{v}}_{k+1}) + \delta_k \Big[f(\mathbf{y}_k) + \big\langle \nabla f(\mathbf{y}_k), \hat{\mathbf{v}}_{k+1} - \mathbf{y}_k \big\rangle \Big] \\ &\overset{(e)}{\geq} (1-\delta_k)\Phi_k^* + \delta_k \Big[f(\mathbf{y}_k) + \big\langle \nabla f(\mathbf{y}_k), \hat{\mathbf{v}}_{k+1} - \mathbf{y}_k \big\rangle \Big] \\ &\overset{(f)}{\geq} (1-\delta_k) f(\mathbf{x}_k) + \delta_k \Big[f(\mathbf{y}_k) + \big\langle \nabla f(\mathbf{y}_k), \hat{\mathbf{v}}_{k+1} - \mathbf{y}_k \big\rangle \Big] - (1-\delta_k)\xi_k \\ &\overset{(g)}{\geq} f(\mathbf{x}_{k+1}) - \frac{L\delta_k^2}{2}\|\hat{\mathbf{v}}_{k+1} - \mathbf{v}_k\|^2 - (1-\delta_k)\xi_k \\ &\geq f(\mathbf{x}_{k+1}) - \frac{LD^2\delta_k^2}{2} - (1-\delta_k)\xi_k, \end{aligned}$$

where (d) is by the definition of $\hat{\Phi}_{k+1}(\mathbf{x})$; (e) uses $\Phi_k(\hat{\mathbf{v}}_{k+1}) \geq \Phi_k^*$; (f) is by the induction hypothesis $f(\mathbf{x}_k) \leq \Phi_k^* + \xi_k$; (g) is by plugging (8.13) in; and the last inequality is because of Assumption 7.3. Rearrange the terms, we have

$$\begin{aligned} f(\mathbf{x}_{k+1}) &\leq \hat{\Phi}_{k+1}^* + \frac{LD^2\delta_k^2}{2} + (1-\delta_k)\xi_k \\ &= \Phi_{k+1}^* + (\hat{\Phi}_{k+1}^* - \Phi_{k+1}^*) + \frac{LD^2\delta_k^2}{2} + (1-\delta_k)\xi_k. \end{aligned} \tag{8.14}$$

Then, we have from Lemma 8.3 that

$$\begin{aligned}
\hat{\Phi}_{k+1}^* - \Phi_{k+1}^* &= s\hat{\Phi}_{k+1}(\hat{\mathbf{v}}_{k+1}) - \Phi_{k+1}(\mathbf{v}_{k+1}) \qquad (8.15)\\
&= \hat{\Phi}_{k+1}(\hat{\mathbf{v}}_{k+1}) - \hat{\Phi}_{k+1}(\mathbf{v}_{k+1}) + \hat{\Phi}_{k+1}(\mathbf{v}_{k+1}) - \Phi_{k+1}(\mathbf{v}_{k+1})\\
&\overset{\text{(h)}}{\leq} \hat{\Phi}_{k+1}(\mathbf{v}_{k+1}) - \Phi_{k+1}(\mathbf{v}_{k+1})\\
&\overset{\text{(i)}}{=} \delta_k \Big[f(\mathbf{y}_k) + \big\langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{y}_k \big\rangle \Big]\\
&\qquad - \delta_k \Big[f(\mathbf{x}_{k+1}) + \big\langle \nabla f(\mathbf{x}_{k+1}), \mathbf{v}_{k+1} - \mathbf{x}_{k+1} \big\rangle \Big]\\
&\overset{\text{(j)}}{\leq} \delta_k \big\langle \nabla f(\mathbf{y}_k) - \nabla f(\mathbf{x}_{k+1}), \mathbf{v}_{k+1} - \mathbf{x}_{k+1} \big\rangle\\
&\leq \delta_k \big\| \nabla f(\mathbf{y}_k) - \nabla f(\mathbf{x}_{k+1}) \big\|_* \big\| \mathbf{v}_{k+1} - \mathbf{x}_{k+1} \big\|\\
&\overset{\text{(k)}}{\leq} \delta_k L \big\| \mathbf{y}_k - \mathbf{x}_{k+1} \big\| \big\| \mathbf{v}_{k+1} - \mathbf{x}_{k+1} \big\|\\
&\overset{\text{(l)}}{\leq} \delta_k^2 L \big\| \mathbf{v}_k - \hat{\mathbf{v}}_{k+1} \big\| \big\| \mathbf{v}_{k+1} - \mathbf{x}_{k+1} \big\| \leq \delta_k^2 L D^2,
\end{aligned}$$

where (h) is because $\hat{\Phi}_{k+1}(\hat{\mathbf{v}}_{k+1}) \leq \hat{\Phi}_{k+1}(\mathbf{x}), \forall \mathbf{x} \in \mathcal{X}$ according to Lemma 8.3; (i) follows from (8.5); (j) uses $f(\mathbf{y}_k) - f(\mathbf{x}_{k+1}) \leq \langle \nabla f(\mathbf{y}_k), \mathbf{y}_k - \mathbf{x}_{k+1} \rangle$; (k) is because of Assumption 7.1; and (l) uses the choice of $\mathbf{y}_k$ and $\mathbf{x}_{k+1}$. Plugging (8.15) back into (8.14), we have

$$f(\mathbf{x}_{k+1}) \leq \Phi_{k+1}^* + \frac{3LD^2\delta_k^2}{2} + (1 - \delta_k)\xi_k,$$

which completes the proof. □

8.4.5 Proof of Theorem 8.1

Proof. Given $\big(\{\Phi_k(\mathbf{x})\}_{k=0}^{\infty}, \{\lambda_k\}_{k=0}^{\infty}\big)$ is an ES as shown in Lemma 8.2, together with the fact $f(\mathbf{x}_k) \leq \min_{\mathbf{x} \in \mathcal{X}} \Phi_k(\mathbf{x}) + \xi_k, \forall k$ as shown in Lemma 8.4, one can directly apply Lemma 8.1 to have

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \lambda_k \big(f(\mathbf{x}_0) - f(\mathbf{x}^*) \big) + \xi_k = \frac{2\big(f(\mathbf{x}_0) - f(\mathbf{x}^*)\big)}{(k+1)(k+2)} + \xi_k, \quad (8.16)$$

where ξ_k is defined in Lemma 8.4. Clearly, $\xi_k \geq 0, \forall k$, and one can find an upper bound of it as

$$\begin{aligned}
\xi_k &= (1-\delta_{k-1})\xi_{k-1} + \frac{3\delta_{k-1}^2}{2} LD^2 \\
&= \frac{3LD^2}{2} \sum_{\tau=0}^{k-1} \delta_\tau^2 \bigg[\prod_{j=\tau+1}^{k-1} (1-\delta_j) \bigg] \\
&= \frac{3LD^2}{2} \sum_{\tau=0}^{k-1} \frac{4}{(\tau+3)^2} \frac{(\tau+2)(\tau+3)}{(k+1)(k+2)} \leq \frac{6LD^2}{k+2}.
\end{aligned}$$

Plugging ξ_k into (8.16) completes the proof. □

8.4.6 Stopping Criterion

In this section, we show that the value of $f(\mathbf{x}_k) - \Phi_k^*$ can be used to derive a stopping criterion (see (8.17)). How to obtain the value of Φ_k^* iteratively (via (8.18) and (8.19)) is also discussed.

First, as a consequence of Lemma 8.4, we have $f(\mathbf{x}_k) - \Phi_k^* \leq \xi_k = \mathcal{O}\big(\frac{LD^2}{k}\big)$. This means that the value of $f(\mathbf{x}_k) - \Phi_k^*$ converges to 0 at the same rate of $f(\mathbf{x}_k) - f(\mathbf{x}^*)$.

Next we show that how to estimate $f(\mathbf{x}_k) - f(\mathbf{x}^*)$ using $f(\mathbf{x}_k) - \Phi_k^*$. We have that

$$\begin{aligned}
f(\mathbf{x}_k) - \Phi_k^* &\overset{\text{(a)}}{\geq} f(\mathbf{x}_k) - \Phi_k(\mathbf{x}^*) \overset{\text{(b)}}{\geq} f(\mathbf{x}_k) - (1-\lambda_k) f(\mathbf{x}^*) - \lambda_k \Phi_0(\mathbf{x}^*) \\
&\overset{\text{(c)}}{=} (1-\lambda_k)\big[f(\mathbf{x}_k) - f(\mathbf{x}^*)\big] + \lambda_k \big[f(\mathbf{x}_k) - f(\mathbf{x}_0)\big],
\end{aligned}$$

where (a) is because of $\Phi_k^* = \min_{\mathbf{x}\in\mathcal{X}} \Phi_k(\mathbf{x})$, (b) is by the definition of ES, and (c) uses $\Phi_0(\mathbf{x}) \equiv f(\mathbf{x}_0)$. The inequality above implies that

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{1}{1-\lambda_k}\Bigg(f(\mathbf{x}_k) - \Phi_k^* - \lambda_k \big[f(\mathbf{x}_k) - f(\mathbf{x}_0)\big]\Bigg). \tag{8.17}$$

Notice that the RHS of (8.17) goes to 0 as k increases, hence (8.17) can be used as the stopping criterion.

Finally we discuss how to update Φ_k^* efficiently. From (8.11), we have

$$\begin{aligned}\Phi_{k+1}(\mathbf{x}) &= f(\mathbf{x}_0)\prod_{\tau=0}^{k}(1-\delta_\tau) + \sum_{\tau=0}^{k}\delta_\tau\Big[f(\mathbf{x}_{\tau+1}) + \langle\nabla f(\mathbf{x}_{\tau+1}), \mathbf{x}-\mathbf{x}_{\tau+1}\rangle\Big] \\ &\qquad \times \prod_{j=\tau+1}^{k}(1-\delta_j) \\ &= f(\mathbf{x}_0)\prod_{\tau=0}^{k}(1-\delta_\tau) + \sum_{\tau=0}^{k}\delta_\tau\Big[f(\mathbf{x}_{\tau+1}) + \langle\nabla f(\mathbf{x}_{\tau+1}), \mathbf{x}-\mathbf{x}_{\tau+1}\rangle\Big] \\ &\qquad \times \prod_{j=\tau+1}^{k}(1-\delta_j) \\ &= f(\mathbf{x}_0)\prod_{\tau=0}^{k}(1-\delta_\tau) + \sum_{\tau=0}^{k}\delta_\tau\Big[f(\mathbf{x}_{\tau+1}) - \langle\nabla f(\mathbf{x}_{\tau+1}), \mathbf{x}_{\tau+1}\rangle\Big] \\ &\qquad \times \prod_{j=\tau+1}^{k}(1-\delta_j) + \langle\mathbf{g}_{k+1}, \mathbf{x}\rangle,\end{aligned}$$

where the last equation uses the definition of $\mathbf{g}_{k+1}$. Hence, we can obtain Φ_{k+1}^* as

$$\Phi_{k+1}^* = \Phi_{k+1}(\mathbf{v}_{k+1}) = V_{k+1} + \langle\mathbf{g}_{k+1}, \mathbf{v}_{k+1}\rangle, \tag{8.18}$$

and V_{k+1} can be updated as

$$\begin{aligned}V_{k+1} = (1-\delta_k)V_k + \delta_k\Big[f(\mathbf{x}_{k+1}) - \langle\nabla f(\mathbf{x}_{k+1}), \mathbf{x}_{k+1}\rangle\Big], \\ \text{with}\ \ V_0 = f(\mathbf{x}_0).\end{aligned} \tag{8.19}$$

8.5 Proof of Theorem 8.2

Because we are dealing with an ℓ_2 norm ball constraint in this section, we use $R := \frac{D}{2}$ for convenience. And we will extend the domain of $f(\mathbf{x})$ slightly to $\tilde{\mathcal{X}} := \text{conv}\{\mathbf{x} - \frac{1}{L}\nabla f(\mathbf{x}),\ \forall \mathbf{x}\in\mathcal{X}\}$, i.e., $f : \tilde{\mathcal{X}} \to \mathbb{R}$. This is a very mild assumption since most of practically used loss functions have domain $\mathbb{R}^d$.

Lemma 8.5. *[125, Theorem 2.1.5] If Assumptions 7.1 and 7.2 hold with the* *extended domain $\tilde{\mathcal{X}}$, then it is true that for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$*

$$\frac{1}{2L}\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2^2 \leq f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle.$$

Lemma 8.6. *Choose $\delta_k = \frac{2}{k+3}$, then we have*

$$\|\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}^*)\|_2 \leq \sqrt{\frac{4L\big(f(\mathbf{x}_0) - f(\mathbf{x}^*)\big)}{(k+1)(k+2)} + \frac{12L^2D^2}{k+2}} \leq \frac{C_1}{\sqrt{k+2}},$$

where $C_1 \leq \sqrt{12L^2D^2 + 4L\big(f(\mathbf{x}_0) - f(\mathbf{x}^*)\big)}$.

Proof. Using Lemma 8.5, we have

$$\begin{aligned}\frac{1}{2L}\|\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}^*)\|_2^2 &\leq f(\mathbf{x}_k) - f(\mathbf{x}^*) - \langle \nabla f(\mathbf{x}^*), \mathbf{x}_k - \mathbf{x}^*\rangle \\ &\overset{\text{(a)}}{\leq} f(\mathbf{x}_k) - f(\mathbf{x}^*) \\ &\overset{\text{(b)}}{\leq} \frac{2\big(f(\mathbf{x}_0) - f(\mathbf{x}^*)\big)}{(k+1)(k+2)} + \frac{6LD^2}{k+2},\end{aligned}$$

where (a) is by the optimality condition, that is, $\langle \nabla f(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^*\rangle \geq 0, \forall \mathbf{x} \in \mathcal{X}$; and (b) is by Theorem 8.1. This further implies

$$\|\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}^*)\|_2 \leq \sqrt{\frac{4L\big(f(\mathbf{x}_0) - f(\mathbf{x}^*)\big)}{(k+1)(k+2)} + \frac{12L^2D^2}{k+2}}.$$

The proof is thus completed. □

Lemma 8.7. *If both $\mathbf{x}_1^*$ and $\mathbf{x}_2^*$ minimize $f(\mathbf{x})$ over $\mathcal{X}$, then we have $\nabla f(\mathbf{x}_1^*) = \nabla f(\mathbf{x}_2^*)$.*

Proof. From Lemma 8.5, we have

$$\begin{aligned}\frac{1}{2L}\|\nabla f(\mathbf{x}_2^*) - \nabla f(\mathbf{x}_1^*)\|_2^2 &\leq f(\mathbf{x}_2^*) - f(\mathbf{x}_1^*) - \langle \nabla f(\mathbf{x}_1^*), \mathbf{x}_2^* - \mathbf{x}_1^*\rangle \\ &\overset{\text{(a)}}{\leq} f(\mathbf{x}_2^*) - f(\mathbf{x}_1^*) = 0,\end{aligned}$$

where (a) is by the optimality condition, that is, $\langle \nabla f(\mathbf{x}_1^*), \mathbf{x} - \mathbf{x}_1^*\rangle \geq 0, \forall \mathbf{x} \in \mathcal{X}$. Hence we can only have $\nabla f(\mathbf{x}_2^*) = \nabla f(\mathbf{x}_1^*)$. This means that the value of $\nabla f(\mathbf{x}^*)$ is unique regardless of the uniqueness of $\mathbf{x}^*$. □

Lemma 8.8. *Let $\|\nabla f(\mathbf{x}^*)\|_2 = G^*$, (and G^* is unique bacause of Lemma 8.7) where $G^* \geq G$. Choose $\delta_k = \frac{2}{k+3}$, it is guaranteed to have*

$$\|\mathbf{g}_{k+1} - \nabla f(\mathbf{x}^*)\|_2 \leq \frac{4C_1}{3(\sqrt{k+3}-1)} + \frac{2G^*}{(k+2)(k+3)}.$$

In addition, there exists a constant $C_2 \leq \frac{4}{3}C_1 + \frac{2}{3(\sqrt{3}+1)}G^*$ such that

$$\|\mathbf{g}_{k+1} - \nabla f(\mathbf{x}^*)\|_2 \leq \frac{C_2}{\sqrt{k+3}-1}.$$

Proof. First we have

$$\begin{aligned}\mathbf{g}_{k+1} = (1-\delta_k)\mathbf{g}_k + \delta_k \nabla f(\mathbf{x}_{k+1}) &= \sum_{\tau=0}^{k} \delta_\tau \nabla f(\mathbf{x}_{\tau+1}) \Big[\prod_{j=\tau+1}^{k}(1-\delta_j)\Big] \qquad (8.20)\\ &= \sum_{\tau=0}^{k} \frac{2(\tau+2)}{(k+2)(k+3)} \nabla f(\mathbf{x}_{\tau+1}).\end{aligned}$$

Noticing that $2\sum_{\tau=0}^{k}(\tau+2) = (k+1)(k+4) = (k+2)(k+3) - 2$, we have

$$\begin{aligned}&\|\mathbf{g}_{k+1} - \nabla f(\mathbf{x}^*)\|_2\\ &= \bigg\| \sum_{\tau=0}^{k} \frac{2(\tau+2)}{(k+2)(k+3)} \big[\nabla f(\mathbf{x}_{\tau+1}) - \nabla f(\mathbf{x}^*)\big] - \frac{2}{(k+2)(k+3)} \nabla f(\mathbf{x}^*) \bigg\|_2\\ &\leq \sum_{\tau=0}^{k} \frac{2(\tau+2)}{(k+2)(k+3)} \big\|\nabla f(\mathbf{x}_{\tau+1}) - \nabla f(\mathbf{x}^*)\big\|_2 + \frac{2}{(k+2)(k+3)} \big\|\nabla f(\mathbf{x}^*)\big\|_2\\ &\overset{(a)}{\leq} \sum_{\tau=0}^{k} \frac{2(\tau+2)}{(k+2)(k+3)} \frac{C_1}{\sqrt{\tau+3}} + \frac{2G^*}{(k+2)(k+3)}\\ &\leq \frac{2C_1}{(k+2)(k+3)} \sum_{\tau=0}^{k} \sqrt{\tau+2} + \frac{2G^*}{(k+2)(k+3)}\\ &\leq \frac{4C_1}{3(k+2)(k+3)} (k+3)^{3/2} + \frac{2G^*}{(k+2)(k+3)}\\ &= \frac{4C_1}{3(\sqrt{k+3}+1)(\sqrt{k+3}-1)} \sqrt{k+3} + \frac{2G^*}{(k+2)(k+3)}\\ &\leq \frac{4C_1}{3(\sqrt{k+3}-1)} + \frac{2G^*}{(k+2)(k+3)},\end{aligned}$$

where (a) follows from Lemma 8.6. This completes the proof for the first part of this lemma. Next, to find C_2, we have

$$\begin{aligned}\|\mathbf{g}_{k+1} - \nabla f(\mathbf{x}^*)\|_2 &\leq \frac{4C_1}{3(\sqrt{k+3}-1)} + \frac{2G^*}{(k+2)(k+3)} \\ &= \frac{4C_1}{3(\sqrt{k+3}-1)} + \frac{2G^*}{(k+3)(\sqrt{k+3}+1)(\sqrt{k+3}-1)} \\ &\overset{\text{(b)}}{\leq} \frac{4C_1}{3(\sqrt{k+3}-1)} + \frac{2G^*}{3(\sqrt{3}+1)(\sqrt{k+3}-1)},\end{aligned}$$

where in (b) we use $k+3 \geq 3$ and $\sqrt{k+3}+1 \geq \sqrt{3}+1$. The proof is thus completed. □

Lemma 8.9. *There exists a constant* $T_1 \leq \big(\frac{2C_2}{G^*}+1\big)^2-3$*, such that* $\|\mathbf{g}_{k+1}\|_2 \geq \frac{G^*}{2}, \forall k \geq T_1$.

Proof. Consider a specific $\tilde{k}$ with $\|\mathbf{g}_{\tilde{k}+1}\|_2 < \frac{G^*}{2}$ satisfied. In this case we have

$$\|\mathbf{g}_{\tilde{k}+1} - \nabla f(\mathbf{x}^*)\|_2 \geq \|\nabla f(\mathbf{x}^*)\|_2 - \|\mathbf{g}_{\tilde{k}+1}\|_2 > G^* - \frac{G^*}{2} = \frac{G^*}{2}.$$

From Lemma 8.8, we have

$$\frac{G^*}{2} < \|\mathbf{g}_{\tilde{k}+1} - \nabla f(\mathbf{x}^*)\|_2 \leq \frac{C_2}{\sqrt{\tilde{k}+3}-1}.$$

From this inequality we can observe that $\|\mathbf{g}_{\tilde{k}+1}\|_2$ can be less than $\frac{\sqrt{G}}{2}$ only when $\tilde{k} < T_1 = \big(\frac{2C_2}{G^*}+1\big)^2 - 3$. Hence, this lemma is proven. □

Lemma 8.10. *Let* $T := \max\{T_1, T_2\}$*, with* $T_2 = \sqrt{\frac{8LD}{G^*}}-3$*. When* $k \geq T+1$*, it is guaranteed that*

$$\|\mathbf{v}_{k+1} - \hat{\mathbf{v}}_{k+1}\|_2 \leq \frac{\delta_k^3 LDC_3}{\|\mathbf{g}_{k+1}\|_2\|\mathbf{g}_k\|_2} \leq \frac{4\delta_k^3 LDC_3}{(G^*)^2}, \tag{8.21}$$

where $C_3 := LD^2 + \frac{DC_2}{\sqrt{2}-1}$.

Proof. First we show that when $k \geq T+1$, both $\|\mathbf{g}_k\|_2 > 0$ and $\|\hat{\mathbf{g}}_{k+1}\|_2 > 0$. First, because $k \geq T+1 \geq T_1+1$, through Lemma 8.9 we have $\|\mathbf{g}_k\|_2 \geq$

$\frac{G^*}{2} > 0$. Then we have

$$\begin{aligned} \big\|\hat{\mathbf{g}}_{k+1}\big\|_2 &= \big\|(1-\delta_k)\mathbf{g}_k + \delta_k \nabla f(\mathbf{x}_{k+1}) - \delta_k \nabla f(\mathbf{x}_{k+1}) + \delta_k \nabla f(\mathbf{y}_k)\big\|_2 \\ &\geq \big\|\mathbf{g}_{k+1}\big\|_2 - \delta_k \big\|\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{y}_k)\big\|_2 \geq \frac{G^*}{2} - \delta_k^2 LD. \end{aligned}$$

The last inequality holds when $k \geq T_1$. Hence when $k \geq \max\{T_1, T_2\} + 1$, we must have both $\|\mathbf{g}_k\|_2 > 0$ and $\|\hat{\mathbf{g}}_{k+1}\|_2 > 0$. Then for any $k \geq T + 1$, in view of (8.6), we can write

$$\begin{aligned} &\|\mathbf{v}_{k+1} - \hat{\mathbf{v}}_{k+1}\|_2 = \left\| - \frac{R}{\|\mathbf{g}_{k+1}\|_2}\mathbf{g}_{k+1} + \frac{R}{\|\hat{\mathbf{g}}_{k+1}\|_2}\hat{\mathbf{g}}_{k+1}\right\|_2 \\ &= \frac{R}{\|\mathbf{g}_{k+1}\|_2\|\hat{\mathbf{g}}_{k+1}\|_2}\left\| \big\|\hat{\mathbf{g}}_{k+1}\big\|_2 \mathbf{g}_{k+1} - \big\|\mathbf{g}_{k+1}\big\|_2 \hat{\mathbf{g}}_{k+1}\right\|_2 \\ &= \frac{R}{\|\mathbf{g}_{k+1}\|_2\|\hat{\mathbf{g}}_{k+1}\|_2} \\ &\qquad \times \left\| \big\|\hat{\mathbf{g}}_{k+1}\big\|_2 \mathbf{g}_{k+1} - \big\|\hat{\mathbf{g}}_{k+1}\big\|_2 \hat{\mathbf{g}}_{k+1} + \big\|\hat{\mathbf{g}}_{k+1}\big\|_2 \hat{\mathbf{g}}_{k+1} - \big\|\mathbf{g}_{k+1}\big\|_2 \hat{\mathbf{g}}_{k+1}\right\|_2 \\ &\leq \frac{R}{\|\mathbf{g}_{k+1}\|_2} \times \left\|\mathbf{g}_{k+1} - \hat{\mathbf{g}}_{k+1}\right\|_2 + \frac{R}{\|\mathbf{g}_{k+1}\|_2}\left| \big\|\hat{\mathbf{g}}_{k+1}\big\|_2 - \big\|\mathbf{g}_{k+1}\big\|_2 \right| \\ &\overset{\text{(a)}}{\leq} \frac{2R}{\|\mathbf{g}_{k+1}\|_2}\left\|\mathbf{g}_{k+1} - \hat{\mathbf{g}}_{k+1}\right\|_2 = \frac{2R\delta_k}{\|\mathbf{g}_{k+1}\|_2}\left\|\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{y}_k)\right\|_2 \\ &\overset{\text{(b)}}{\leq} \frac{2RL\delta_k}{\|\mathbf{g}_{k+1}\|_2}\left\|\mathbf{x}_{k+1} - \mathbf{y}_k\right\|_2 = \frac{DL\delta_k^2}{\|\mathbf{g}_{k+1}\|_2}\left\|\hat{\mathbf{v}}_{k+1} - \mathbf{v}_k\right\|_2, \end{aligned} \tag{8.22}$$

where (a) is by $\big|\|\mathbf{a}\|_2 - \|\mathbf{b}\|_2\big| \leq \big\|\mathbf{a} - \mathbf{b}\big\|_2$; and (b) is by Assumption 7.1. Then we will bound $\|\hat{\mathbf{v}}_{k+1} - \mathbf{v}_k\|_2$.

$$\begin{aligned} &\big\|\hat{\mathbf{v}}_{k+1} - \mathbf{v}_k\big\|_2 = \left\| - \frac{R}{\|\hat{\mathbf{g}}_{k+1}\|_2}\hat{\mathbf{g}}_{k+1} + \frac{R}{\|\mathbf{g}_k\|_2}\mathbf{g}_k\right\|_2 \\ &= \frac{R}{\|\mathbf{g}_k\|_2\|\hat{\mathbf{g}}_{k+1}\|_2} \\ &\qquad \times \left\| \big\|\mathbf{g}_k\big\|_2 \hat{\mathbf{g}}_{k+1} - \big\|\hat{\mathbf{g}}_{k+1}\big\|_2 \hat{\mathbf{g}}_{k+1} + \big\|\hat{\mathbf{g}}_{k+1}\big\|_2 \hat{\mathbf{g}}_{k+1} - \big\|\hat{\mathbf{g}}_{k+1}\big\|_2 \mathbf{g}_k\right\|_2 \\ &\leq \frac{R}{\|\mathbf{g}_k\|_2}\left| \big\|\mathbf{g}_k\big\|_2 - \big\|\hat{\mathbf{g}}_{k+1}\big\|_2 \right| + \frac{R}{\|\mathbf{g}_k\|_2}\left\|\hat{\mathbf{g}}_{k+1} - \mathbf{g}_k\right\|_2 \\ &\overset{\text{(c)}}{\leq} \frac{D}{\|\mathbf{g}_k\|_2}\left\|\hat{\mathbf{g}}_{k+1} - \mathbf{g}_k\right\|_2 = \frac{\delta_k D}{\|\mathbf{g}_k\|_2}\left\|\nabla f(\mathbf{y}_k) - \mathbf{g}_k\right\|_2 \end{aligned}$$

$$
\begin{aligned}
&\leq \frac{\delta_k D}{\|\mathbf{g}_k\|_2}\big\|\nabla f(\mathbf{y}_k) - \nabla f(\mathbf{x}^*)\big\|_2 + \frac{\delta_k D}{\|\mathbf{g}_k\|_2}\big\|\nabla f(\mathbf{x}^*) - \mathbf{g}_k\big\|_2 \\
&\leq \frac{\delta_k L D^2}{\|\mathbf{g}_k\|_2} + \frac{\delta_k D}{\|\mathbf{g}_k\|_2}\big\|\nabla f(\mathbf{x}^*) - \mathbf{g}_k\big\|_2 \\
&\leq \frac{\delta_k L D^2}{\|\mathbf{g}_k\|_2} + \frac{\delta_k D}{\|\mathbf{g}_k\|_2}\frac{C_2}{\sqrt{k+2}-1} \leq \frac{\delta_k\big(LD^2 + \frac{DC_2}{\sqrt{T+3}-1}\big)}{\|\mathbf{g}_k\|_2} := \frac{\delta_k C_3}{\|\mathbf{g}_k\|_2},
\end{aligned}
$$

where (c) again uses $\big|\|\mathbf{a}\|_2 - \|\mathbf{b}|_2\big| \leq \big\|\mathbf{a}-\mathbf{b}\big\|_2$, and the last inequality is because of Lemma 8.6. Plugging back to (8.22), we arrive at

$$
\|\mathbf{v}_{k+1} - \hat{\mathbf{v}}_{k+1}\|_2 \leq \frac{DL\delta_k^2}{\|\mathbf{g}_{k+1}\|_2}\frac{\delta_k C_3}{\|\mathbf{g}_k\|_2} = \frac{\delta_k^3 LDC_3}{\|\mathbf{g}_{k+1}\|_2\|\mathbf{g}_k\|_2} \leq \frac{4\delta_k^3 LDC_3}{(G^*)^2}.
$$

The proof is thus completed. □

Lemma 8.11. *Let $\xi_0 = 0$ and T defined the same as in Lemma 8.10. Denote $\Phi_k^* := \Phi_k(\mathbf{v}_k)$ as the minimum value of $\Phi_k(\mathbf{x})$ over $\mathcal{X}$, then we have*

$$
f(\mathbf{x}_k) \leq \Phi_k^* + \xi_k, \forall k \geq 0,
$$

where for $k < T+1$, $\xi_{k+1} = (1-\delta_k)\xi_k + \frac{3LD^2}{2}\delta_k^2$, and $\xi_{k+1} = C_4\delta_k^4 + (1-\delta_k)\xi_k$ for $k \geq T+1$ with $C_4 = \Big(\frac{C_1}{\sqrt{T+4}} + G^*\Big)\frac{4LDC_3}{(G^*)^2}$.

Proof. The proof for $k < T+1$ is similar as that in Lemma 8.4, hence it is omitted here. We mainly focus on the case where $k \geq T+1$.

$$
\begin{aligned}
\Phi_{k+1}^* &= \Phi_{k+1}(\mathbf{v}_{k+1}) \\
&= (1-\delta_k)\Phi_k(\mathbf{v}_{k+1}) + \delta_k\Big[f(\mathbf{x}_{k+1}) + \big\langle\nabla f(\mathbf{x}_{k+1}), \mathbf{v}_{k+1} - \mathbf{x}_{k+1}\big\rangle\Big] \\
&\overset{(a)}{\geq} (1-\delta_k)\Phi_k(\mathbf{v}_k) + \delta_k\Big[f(\mathbf{x}_{k+1}) + \big\langle\nabla f(\mathbf{x}_{k+1}), \mathbf{v}_{k+1} - \mathbf{x}_{k+1}\big\rangle\Big] \\
&\geq (1-\delta_k)f(\mathbf{x}_k) + \delta_k\Big[f(\mathbf{x}_{k+1}) + \big\langle\nabla f(\mathbf{x}_{k+1}), \mathbf{v}_{k+1} - \mathbf{x}_{k+1}\big\rangle\Big] \\
&\qquad - (1-\delta_k)\xi_k \\
&= f(\mathbf{x}_{k+1}) + (1-\delta_k)\big[f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})\big] \\
&\qquad + \delta_k\big\langle\nabla f(\mathbf{x}_{k+1}), \mathbf{v}_{k+1} - \mathbf{x}_{k+1}\big\rangle - (1-\delta_k)\xi_k \\
&\overset{(b)}{\geq} f(\mathbf{x}_{k+1}) + (1-\delta_k)\big\langle\nabla f(\mathbf{x}_{k+1}), \mathbf{x}_k - \mathbf{x}_{k+1}\big\rangle \\
&\qquad + \delta_k\big\langle\nabla f(\mathbf{x}_{k+1}), \mathbf{v}_{k+1} - \mathbf{x}_{k+1}\big\rangle - (1-\delta_k)\xi_k
\end{aligned}
$$

$$
\begin{aligned}
&= f(\mathbf{x}_{k+1}) + \delta_k \big\langle \nabla f(\mathbf{x}_{k+1}), \mathbf{v}_{k+1} - \hat{\mathbf{v}}_{k+1} \big\rangle - (1-\delta_k)\xi_k \\
&\overset{(c)}{\geq} f(\mathbf{x}_{k+1}) - \delta_k \|\nabla f(\mathbf{x}_{k+1})\|_2 \|\mathbf{v}_{k+1} - \hat{\mathbf{v}}_{k+1}\|_2 - (1-\delta_k)\xi_k \\
&\overset{(d)}{\geq} f(\mathbf{x}_{k+1}) - \|\nabla f(\mathbf{x}_{k+1})\|_2 \frac{4\delta_k^4 LDC_3}{(G^*)^2} - (1-\delta_k)\xi_k \\
&\overset{(e)}{\geq} f(\mathbf{x}_{k+1}) - \Big(\frac{C_1}{\sqrt{T+4}} + G^* \Big) \frac{4\delta_k^4 LDC_3}{(G^*)^2} - (1-\delta_k)\xi_k,
\end{aligned}
$$

where (a) is because $\mathbf{v}_k$ minimizes $\Phi_k(\mathbf{x})$ shown in Lemma 8.3; (b) is by $f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) \leq \langle \nabla f(\mathbf{x}_{k+1}), \mathbf{x}_{k+1} - \mathbf{x}_k \rangle$; (c) uses Cauchy-Schwarz inequality; (d) uses Lemma 8.10, and (e) uses the following inequality.

$$
\begin{aligned}
\|\nabla f(\mathbf{x}_{k+1})\|_2 &= \|\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)\|_2 \\
&\leq \|\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}^*)\|_2 + \|\nabla f(\mathbf{x}^*)\|_2 \\
&\leq \frac{C_1}{\sqrt{k+3}} + G^* \leq \frac{C_1}{\sqrt{T+4}} + G^*,
\end{aligned}
$$

where the last line uses Lemma 8.6. □

Proof of Theorem 8.2

Proof. Let T be defined the same as in Lemma 8.9. For convenience denote $\xi_{k+1} = (1-\delta_k)\xi_k + \theta_k$. When $k < T+1$, we have $\theta_k = \frac{3LD^2}{2}\delta_k^2$; when $k \geq T+1$, we have $\theta_k = C_4 \delta_k^4$.

Then we can write

$$
\begin{aligned}
\xi_{k+1} = (1-\delta_k)\xi_k + \theta_k &= \sum_{\tau=0}^{k} \theta_\tau \prod_{j=\tau+1}^{k} (1-\delta_j) \\
&= \sum_{\tau=0}^{k} \theta_\tau \frac{(\tau+2)(\tau+3)}{(k+2)(k+3)} \\
&= \sum_{\tau=0}^{T} \frac{3LD^2}{2} \delta_\tau^2 \frac{(\tau+2)(\tau+3)}{(k+2)(k+3)} + \sum_{\tau=T+1}^{k} C_4 \delta_\tau^4 \frac{(\tau+2)(\tau+3)}{(k+2)(k+3)} \\
&= \frac{6LD^2(T+1)}{(k+2)(k+3)} + \mathcal{O}\Big(\frac{C_4}{k^3} \Big). \qquad (8.23)
\end{aligned}
$$

Again note that $T < \mathcal{O}\big(\max\{ \sqrt{\frac{LD}{G}}, \frac{L^2D^2}{G^2} \} \big)$ is a constant independent of

k. Finally, applying Lemma 8.1, we have

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{2\big[f(\mathbf{x}_0) - f(\mathbf{x}^*)\big]}{(k+1)(k+2)} + \xi_k. \tag{8.24}$$

Plugging the expression of ξ_k, i.e., (8.23), into (8.24) completes the proof. □

8.6 Discussions for Other Constraints

8.6.1 ℓ_1 norm ball

In this section we focus on the convergence of ExtraFW for ℓ_1 norm ball constraint under the assumption that $\arg\max_j \big|[\nabla f(\mathbf{x}^*)]_j\big|$ has cardinality 1 (which is also known as *strict complementarity* [246], and it naturally implies that the constraint is active). Note that in this case Lemma 8.7 still holds, hence the value of $\nabla f(\mathbf{x}^*)$ is unique regardless the uniqueness of $\mathbf{x}^*$. This assumption directly leads to $\arg\max_j \big|[\nabla f(\mathbf{x}^*)]_j\big| - |[\nabla f(\mathbf{x}^*)]_i| \geq \lambda, \forall i$ for some $\lambda > 0$.

The closed-form solution of $\mathbf{v}_{k+1}$ is given in (8.7). The constants required in the proof are summarized below for clearance. The norm considered in this section for defining L and D is $\|\cdot\|_1$; that is, $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_\infty \leq L\|\mathbf{x}-\mathbf{y}\|_1$, and $\|\mathbf{x}-\mathbf{y}\|_1 \leq D, \forall \mathbf{x}, \forall \mathbf{y} \in \tilde{\mathcal{X}}$. Using equivalences of norms, we also assume $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq L_2\|\mathbf{x}-\mathbf{y}\|_2, \forall \mathbf{x}, \mathbf{y} \in \tilde{\mathcal{X}}$, and $\|\mathbf{x}-\mathbf{y}\|_2 \leq D_2, \forall \mathbf{x}, \forall \mathbf{y} \in \mathcal{X}$.

Lemma 8.12. *There exists a constant T (which is irreverent with k), whenever $k \geq T$, it is guaranteed to have*

$$\|\mathbf{v}_{k+1} - \hat{\mathbf{v}}_{k+1}\|_1 = 0.$$

Proof. In the proof, we denote $i = \arg\max_j |[\nabla f(\mathbf{x}^*)]_j|$ for convenience. With $\|\nabla f(\mathbf{x}^*)\|_2 = G^*$, Lemma 8.8 still holds.

We first show that there exists $T_1 = (\frac{3C_2}{\lambda} + 1)^2 - 3$, such that for all $k \geq T_1$, we have $\arg\max_j |[\mathbf{g}_{k+1}]_j| = i$, which further implies only the i-th entry of $\mathbf{v}_{k+1}$ is non-zero. Since Lemma 8.8 holds, one can see whenever $k \geq T_1$, it is guaranteed to have $\|\mathbf{g}_{k+1} - \nabla f(\mathbf{x}^*)\|_2 \leq \frac{\lambda}{3}$. Therefore, one must have $\big||[\mathbf{g}_{k+1}]_j| - |[\nabla f(\mathbf{x}^*)]_j|\big| \leq \frac{\lambda}{3}, \forall j$. Then it is easy to see that $|[\mathbf{g}_{k+1}]_i| - |[\mathbf{g}_{k+1}]_j| \geq \frac{\lambda}{3}, \forall j$. Hence, we have $\arg\max_j |[\mathbf{g}_{k+1}]_j| = i$.

Next we show that there exists another constant $T = \max\{T_1, (\frac{3C_5}{\lambda})^2 - 3\}$, such that $\arg\max_j |[\hat{\mathbf{g}}_{k+1}]_j| = i, \forall k \geq T$, which further indicates only the i-th entry of $\hat{\mathbf{v}}_{k+1}$ is non-zero. In this case, in view of Lemma 8.8, we have

$$\begin{aligned}
&\big\|\hat{\mathbf{g}}_{k+1} - \nabla f(\mathbf{x}^*)\big\|_2 \\
&= \big\|(1-\delta_k)\mathbf{g}_k + \delta_k \nabla f(\mathbf{x}_{k+1}) - \delta_k \nabla f(\mathbf{x}_{k+1}) + \delta_k \nabla f(\mathbf{y}_k) - \nabla f(\mathbf{x}^*)\big\|_2 \\
&\leq \big\|\mathbf{g}_{k+1} - \nabla f(\mathbf{x}^*)\|_2 + \delta_k \|\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{y}_k)\big\|_2 \\
&\leq \big\|\mathbf{g}_{k+1} - \nabla f(\mathbf{x}^*)\|_2 + \delta_k^2 L_2 D_2 \\
&\leq \frac{C_2}{\sqrt{k+3}-1} + \frac{4L_2D_2}{(k+3)^2} \leq \frac{C_5}{\sqrt{k+3}-1}, \forall k \geq T_1,
\end{aligned}$$

where $C_5 \leq C_2 + \frac{4L_2D_2}{(\sqrt{T_1+3}-1)^3}$.

Hence whenever $k \geq \max\{T_1, (\frac{3C_5}{\lambda}+1)^2 - 3\}$, it is guaranteed to have $\|\hat{\mathbf{g}}_{k+1} - \nabla f(\mathbf{x}^*)\|_2 \leq \frac{\lambda}{3}$. Therefore, one must have $\big||[\hat{\mathbf{g}}_{k+1}]_j| - |[\nabla f(\mathbf{x}^*)]_j|\big| \leq \frac{\lambda}{3}, \forall j$. It is thus straightforward to see that $|[\hat{\mathbf{g}}_{k+1}]_i| - |[\hat{\mathbf{g}}_{k+1}]_j| \geq \frac{\lambda}{3}, \forall j$. Hence, it is clear that $\arg\max_j |[\hat{\mathbf{g}}_{k+1}]_j| = i$.

Then one can see that when $k \geq T$, we have $\mathbf{v}_{k+1} - \hat{\mathbf{v}}_{k+1} = \mathbf{0}$. □

Next, we modify Lemma 8.11 to cope with the ℓ_1 norm ball constraint.

Lemma 8.13. *Let $\xi_0 = 0$ and T be the same as in Lemma 8.12. Denote $\Phi_k^* := \Phi_k(\mathbf{v}_k)$ as the minimum value of $\Phi_k(\mathbf{x})$ over $\mathcal{X}$, then we have*

$$f(\mathbf{x}_k) \leq \Phi_k(\mathbf{v}_k) = \Phi_k^* + \xi_k, \forall k \geq 0,$$

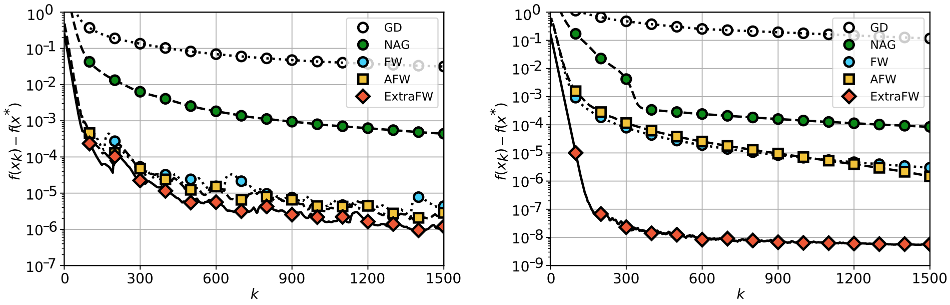
where for $k < T$, $\xi_{k+1} = (1-\delta_k)\xi_k + \frac{3LD^2}{2}\delta_k^2$, and $\xi_{k+1} = (1-\delta_k)\xi_k$ for $k \geq T$.

Proof. The proof for $k < T$ is similar as that in Lemma 8.4, hence it is omitted here. We mainly focus on the case where $k \geq T$. Using similar argument as in Lemma 8.11, we have

$$\begin{aligned}
\Phi_{k+1}^* &\geq f(\mathbf{x}_{k+1}) + \delta_k \big\langle \nabla f(\mathbf{x}_{k+1}), \mathbf{v}_{k+1} - \hat{\mathbf{v}}_{k+1}\big\rangle - (1-\delta_k)\xi_k \\
&= f(\mathbf{x}_{k+1}) - (1-\delta_k)\xi_k,
\end{aligned}$$

where the last inequality is because of Lemma 8.12. □

Theorem 8.3. *Consider $\mathcal{X}$ is an ℓ_1 norm ball. If $\arg\max_j \big|[\nabla f(\mathbf{x}^*)]_j\big|$ has cardinality 1, and Assumptions 7.1 - 7.3 are satisfied, ExtraFW guarantees*



(a) *mnist* (b) *mushroom*

Figure 8.5: ExtraFW guarantees an $\mathcal{O}(\frac{1}{k^2})$ rate on simplex.

that

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) = \mathcal{O}\Big(\frac{1}{k^2}\Big).$$

Proof. Let T be defined the same as in Lemma 8.12. For convenience, denote $\xi_{k+1} = (1-\delta_k)\xi_k + \theta_k$. When $k < T$, we have $\theta_k = \frac{3LD^2}{2}\delta_k^2$; when $k \geq T$, we have $\theta_k = 0$. Then we can write

$$\begin{aligned}\xi_{k+1} = (1-\delta_k)\xi_k + \theta_k &= \sum_{\tau=0}^{k} \theta_\tau \prod_{j=\tau+1}^{k} (1-\delta_j) = \sum_{\tau=0}^{k} \theta_\tau \frac{(\tau+2)(\tau+3)}{(k+2)(k+3)} \\ &= \sum_{\tau=0}^{T-1} \frac{3LD^2}{2}\delta_\tau^2 \frac{(\tau+2)(\tau+3)}{(k+2)(k+3)} = \frac{6LD^2T}{(k+2)(k+3)}. \end{aligned} \tag{8.25}$$

Finally, applying Lemma 8.1, we have

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{2\big[f(\mathbf{x}_0) - f(\mathbf{x}^*)\big]}{(k+1)(k+2)} + \xi_k. \tag{8.26}$$

Plugging the expression of ξ_k, i.e., (8.25) into (8.26) completes the proof. □

Beyond ℓ_1 norm ball. The $\mathcal{O}(\frac{T}{k^2})$ rate in Theorem 8.3 can be generalized in a straightforward manner to simplex, that is, $\mathcal{X} := \{\mathbf{x} | \mathbf{x} \geq \mathbf{0}, \langle \mathbf{1}, \mathbf{x} \rangle = R\}$ for some $R > 0$. A minor assumption needed is that the cardinality of $\arg\min_j [\nabla f(\mathbf{x}^*)]_j$ is 1. In this case, the FW steps in ExtraFW admit closed-form solutions. Again taking $\mathbf{v}_{k+1}$ as an example, we have $\mathbf{v}_{k+1} = [0, \ldots, 0, R, 0, \ldots, 0]$, where the only non-zero is the $i = \arg\min_j [\mathbf{g}_{k+1}]_j$-th entry. The proof is similar to the ℓ_1 norm ball case, i.e., first show that both $\mathbf{g}_{k+1}$ and $\hat{\mathbf{g}}_{k+1}$ converge to $\nabla f(\mathbf{x}^*)$ so that $\mathbf{v}_{k+1} = \hat{\mathbf{v}}_{k+1}, \forall k \geq T$, where T

is some constant depending on the difference of the smallest and the second smallest entry of $\nabla f(\mathbf{x}^*)$. Then one can follow similar steps of Lemma 8.13 to obtain the $\mathcal{O}(\frac{T}{k^2})$ rate. Numerical evidences using logistic regression as objective function can be found in Fig. 8.5. Note that in this case however, FW itself converges fast enough.

8.6.2 n-support norm ball

When $\mathcal{X}$ is an n-support norm ball, ExtraFW guarantees that $f(\mathbf{x}_k) - f(\mathbf{x}^*) = \mathcal{O}\big(\frac{T}{k^2}\big)$. The proof is just a combination of Theorem 8.2 and 8.3: therefore, we highlight the general idea rather than repeat the proofs step by step.

The norm considered in this section for defining L and D is $\|\cdot\|_2$, that is, $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq L\|\mathbf{x}-\mathbf{y}\|_2, \forall \mathbf{x},\mathbf{y} \in \tilde{\mathcal{X}}$, and $\|\mathbf{x}-\mathbf{y}\|_2 \leq D, \forall \mathbf{x},\mathbf{y} \in \mathcal{X}$. Besides Assumptions 7.1 - 7.3, the extra regularity condition we need is that the n-th largest entry of $|[\nabla f(\mathbf{x}^*)]|$ is strictly larger than the $(n+1)$-th largest entry of $|[\nabla f(\mathbf{x}^*)]|$ by λ. Note that this condition is similar to what we used for the ℓ_1 norm ball constraint. In addition, this extra assumption directly implies $\|\nabla f(\mathbf{x}^*)\|_2 := G^* > 0$. In the proof one may find the constant $G_n^* := \|\mathrm{top}_n(\nabla f(\mathbf{x}^*))\|_2$ helpful. Clearly, $G^* \geq G_n^* \geq \sqrt{\frac{n}{d}}G^*$.

Theorem 8.4. *Consider $\mathcal{X}$ is an n-support norm ball. If the n-th largest entry of $|[\nabla f(\mathbf{x}^*)]|$ is strictly larger than the $(n+1)$-th largest entry of $|[\nabla f(\mathbf{x}^*)]|$ and Assumptions 7.1 - 7.3 are satisfied, ExtraFW guarantees that there exists a constant T such that*

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) = \mathcal{O}\Big(\frac{T}{k^2}\Big).$$

Proof. First by using the regularity condition and similar arguments of Lemma 8.12, one can show that there exists a constant T_1 (depending on λ, L, D, and G) such that the indices of the non-zero entries of $\mathbf{v}_{k+1}$ and $\hat{\mathbf{v}}_{k+1}$ are the same for all $k \geq T_1$.

Next, using similar arguments of Lemma 8.9, one can show that there exists a constant $\tilde{T}_2$ such that $\|\mathrm{top}_n(\mathbf{g}_{k+1})\|_2 \geq \frac{G_n^*}{2}$.

Let $T_2 = \max\{\tilde{T}_2, T_1\}$. It is clear that for any $k \geq T_2$, the indices of non-zero entries of $\mathbf{v}_{k+1}$ and $\hat{\mathbf{v}}_{k+1}$ are the same. Together with $\|\mathrm{top}_n(\mathbf{g}_{k+1})\|_2 \geq$

$\frac{G_n^*}{2}, \forall k \geq T_2$, we can show that for any $k \geq T_2 + 1$, $\|\mathbf{v}_{k+1} - \hat{\mathbf{v}}_{k+1}\|_2 = \mathcal{O}(\delta_k^3)$ holds through similar steps as Lemma 8.10.

Finally, using similar arguments of Lemma 8.11 with the aid of $\|\mathbf{v}_{k+1} - \hat{\mathbf{v}}_{k+1}\|_2 = \mathcal{O}(\delta_k^3)$, and applying Lemma 8.1, we can obtain $f(\mathbf{x}_k) - f(\mathbf{x}^*) = \mathcal{O}\big(\frac{T_2}{k^2}\big)$. □

8.7 Additional Numerical Results

8.7.1 Efficiency of ExtraFW: Case Study of n-support Norm Ball

In this section we show that ExtraFW achieves fast convergence rate and low iteration cost simultaneously when the constraint set is an n-support norm ball. We compare algorithms that can solve the constrained formulation or its equivalent regularized formulation discussed in Section 8.2.3, that is

$$\min_{\mathbf{x}} \; f(\mathbf{x}) + \lambda(\|\mathbf{x}\|_{\text{n}-\text{sp}})^2 \tag{8.27a}$$

$$\Leftrightarrow \quad \min_{\mathbf{x}} \; f(\mathbf{x}) \;\; \text{s.t.} \;\; \|\mathbf{x}\|_{\text{n}-\text{sp}} \leq R, \tag{8.27b}$$

where $\|\cdot\|_{\text{n}-\text{sp}}$ denotes the n-support norm [244].

Clearly, one can apply proximal NAG (Prox-NAG) to (8.27a). The proximal operator per iteration has complexity $\mathcal{O}(d(n + \log d))$ [244].

One can also apply ExtraFW for (8.27b). From the Lagrangian duality of (8.27b) and (8.27a), one can see that if $\lambda \neq 0$, one must have an optimal solution for (8.27b) lies on the boundary of its constraint set. Hence ExtraFW achieves acceleration in this case. Below we summarize the convergence rate and per iteration cost of different algorithms. A simple comparison among different algorithms illustrates the efficiency of ExtraFW.

Table 8.1: A comparison of different algorithms for logistic regression with n-support norm

Algorithm	convergence rate	per iteration cost
Prox-NAG for (8.27a)	$\mathcal{O}(1/k^2)$	proximal operator: $\mathcal{O}(d(n + \log d))$
Projected NAG for (8.27b)	$\mathcal{O}(1/k^2)$	projection is expensive
FW for (8.27b)	$\mathcal{O}(1/k)$	FW step: $\mathcal{O}(d \log n)$
ExtraFW for (8.27b)	$\mathcal{O}(T/k^2)$	FW step: $\mathcal{O}(d \log n)$

8.7.2 Binary Classification

Table 8.2: A summary of datasets used in numerical experiments

Dataset	d	N (train)	nonzeros
w7a	300	$24,692$	3.89%
realsim	$20,958$	$50,617$	0.24%
news20	$19,996$	$1,355,191$	0.033%
mushromm	122	$8,124$	18.75%
mnist (digit 4)	784	$60,000$	12.4%

The datasets used for the experiments are summarized in Table 8.2.

Sparsity promoting property of FW variants in ℓ_1 norm ball constraint. FW in Algorithm 8.1 directly promotes sparsity on the solution if it is initialized at $\mathbf{x}_0 = \mathbf{0}$. To see this, suppose that the i-th entry of $\nabla f(\mathbf{x}_k)$ has the largest absolute value, then we have $\mathbf{v}_{k+1} = [0, \ldots, -\text{sgn}\big([\nabla f(\mathbf{x}_k)]_i\big) R, \ldots, 0]^\top$ with the i-th entry being non-zero. Hence, $\mathbf{x}_k$ has at most k non-zero entries given $k-1$ entries are non-zero in $\mathbf{x}_{k-1}$. This sparsity promoting property also holds for ExtraFW.

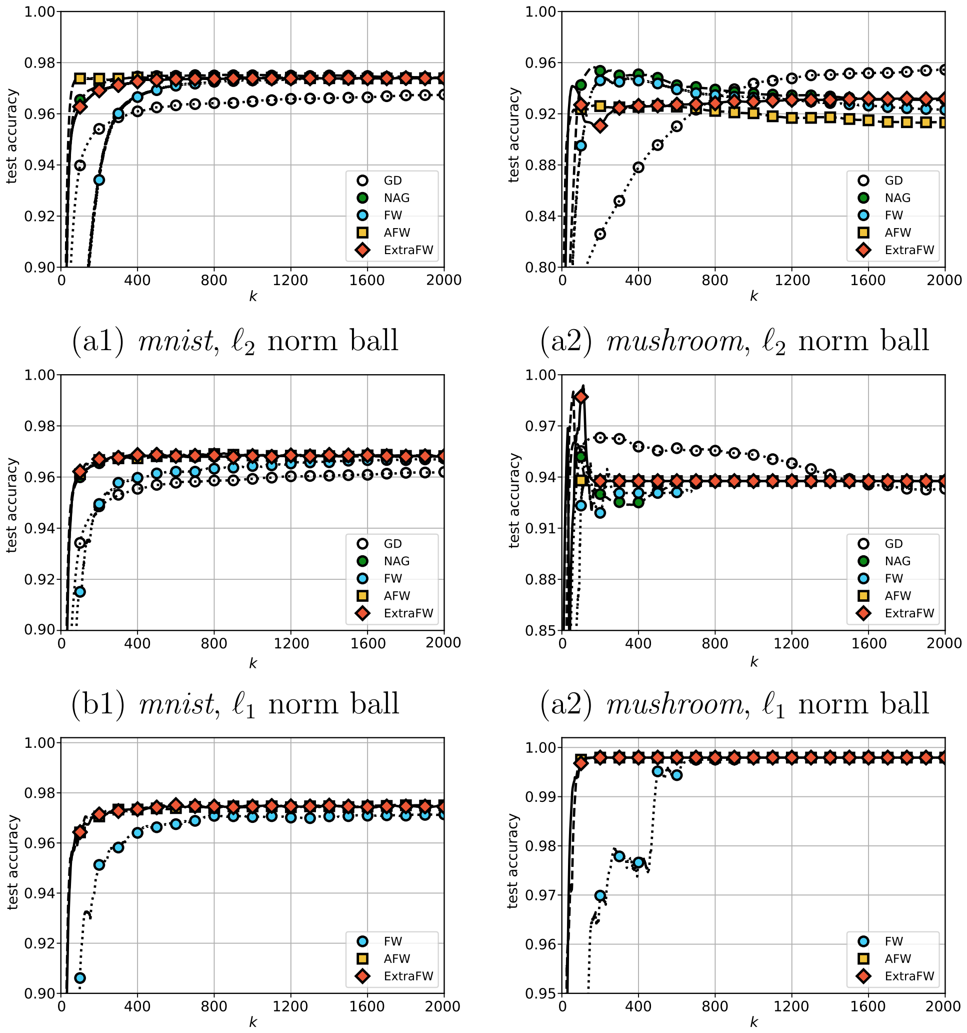
The experiment accuracy of different algorithms can be found in Fig. 8.6. Additional numerical results for ℓ_1 norm ball constraint can be found in Fig. 8.7. It can be seen that on dataset *realsim*, ExtraFW has similar performance with AFW, both outperforming FW significantly. On dataset *news20*, ExtraFW outperforms AFW in terms of optimality error.

Additional experiments for n-support norm ball constraint are listed in Fig. 8.8. The optimality error of ExtraFW is smaller than AFW on both *realsim* and *news20*.

8.7.3 Matrix Completion

Besides the projection-free property, FW and its variants are more suitable for problem (8.10) compared to GD/NAG because they also guarantee $\text{rank}(\mathbf{X}_k) \leq k+1$ [143, 142]. Take FW in Algorithm 8.1 for example. First it is clear that $\nabla f(\mathbf{X}_k) = (\mathbf{X}_k - \mathbf{A})_{\mathcal{K}}$. Suppose the SVD of $\nabla f(\mathbf{X}_k)$ is given by $\nabla f(\mathbf{X}_k) = \mathbf{P}_k \mathbf{\Sigma}_k \mathbf{Q}_k^\top$. Then the FW step can be solved easily by

$$\boldsymbol{V}_{k+1} = -R\mathbf{p}_k\mathbf{q}_k^\top, \tag{8.28}$$



(a1) *mnist*, ℓ_2 norm ball

(a2) *mushroom*, ℓ_2 norm ball

(b1) *mnist*, ℓ_1 norm ball

(a2) *mushroom*, ℓ_1 norm ball

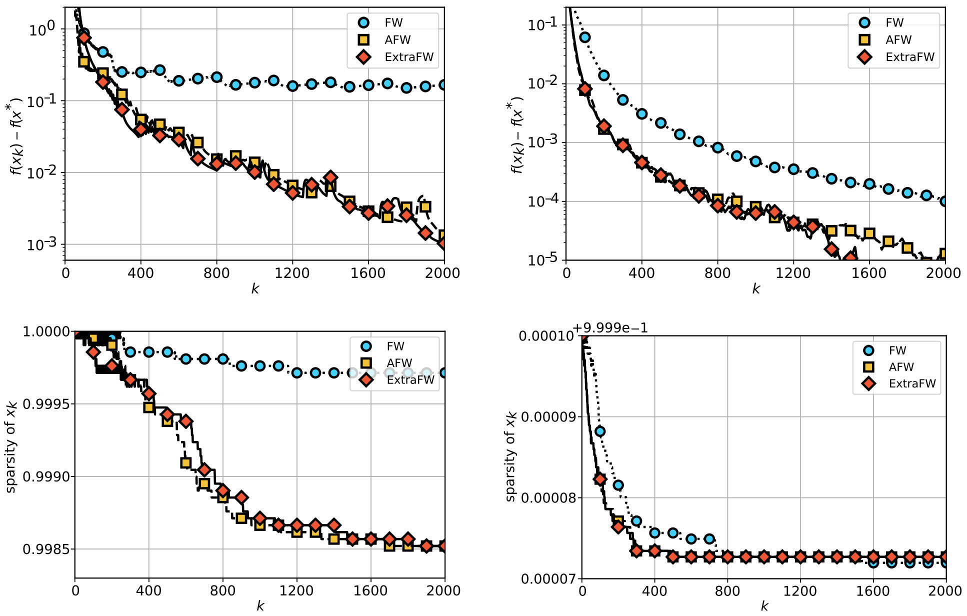
(c1) *mnist*, n-supp norm ball

(c2) *mushroom*, n-supp norm ball

Figure 8.6: Experiment accuracy of ExtraFW on different constraints.

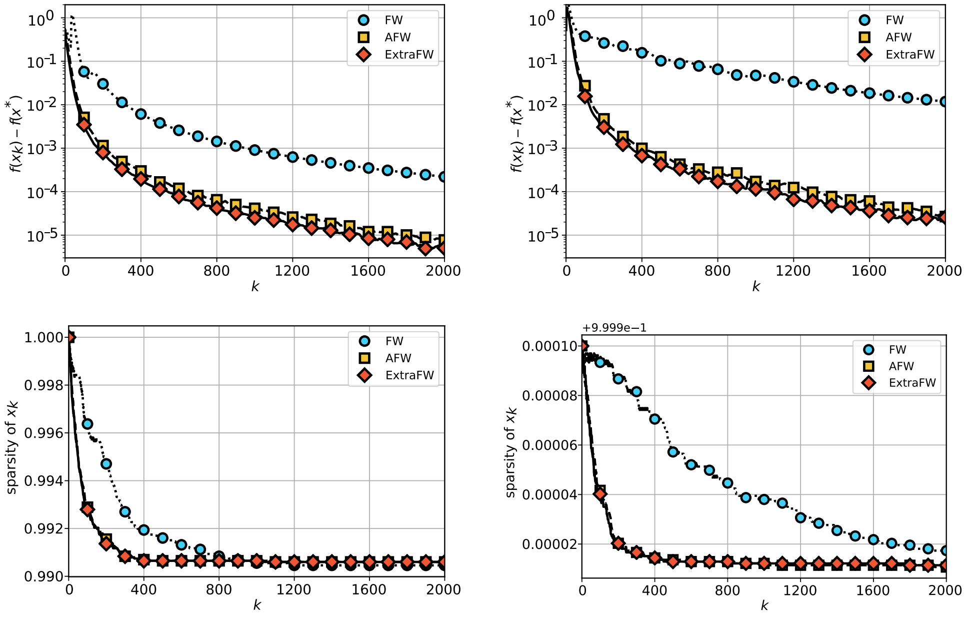
where $\mathbf{p}_k$ and $\mathbf{q}_k$ denote the left and right singular vectors corresponding to the largest singular value of $\nabla f(\mathbf{X}_k)$, respectively. Clearly $\boldsymbol{V}_{k+1}$ in (8.28) has rank at most 1. Hence it is easy to see that $\mathbf{X}_{k+1} = (1-\delta_k)\mathbf{X}_k + \delta_k \boldsymbol{V}_{k+1}$ has rank at most $k+2$ if $\mathbf{X}_k$ is a rank-$(k+1)$ matrix (i.e., $\mathbf{X}_0$ has rank 1). Using similar arguments, ExtraFW also ensures $\text{rank}(\mathbf{X}_k) \leq k+1$. Therefore, the low rank structure is directly promoted by FW variants, and a faster convergence in this case implies a guaranteed lower rank $\mathbf{X}_k$.

The dataset used for the experiment is *MovieLens100K*, where 1682 movies are rated by 943 users with 6.30% percent ratings observed. The initialization and data processing are the same as those used in [142].



(a) *realsim* (b) *news20*

Figure 8.7: Additional experiments of ExtraFW for classification with $\mathcal{X}$ being an ℓ_1 norm ball.



(a) *realsim* (b) *news20*

Figure 8.8: Additional experiments of ExtraFW for classification with $\mathcal{X}$ being an n-support norm ball.

CHAPTER 9

CONCLUSION

Chapter 3 proposes a robust DF framework with backtest-based bootstrap and adaptive residual selection. It can efficiently extend an arbitrary PF model to generate DF, is robust to the choice of model, and outperforms a variety of benchmark DF methods on real-world data, making the proposed framework well-suited for industrial applications.

Chapter 2 addresses the ENSO region spatio-temporal sequence prediction problem by proposing a modified ConvGRU network, as well as its downstream task of predicting the Niño 3.4 index. The ConvGRU network incorporated 2-D convolutional layers within a ConvGRU cell and employed an encoder-decoder Seq2Seq structure, offering advantages over existing models such as LR, LIMs, CNN, KAF, and Seq2Seq with GRU. These advantages include the ability to output future SST maps of the ENSO region, rather than ENSO indices, and modelling approximate nonlinear dynamics. Through experiments on various climate and atmospheric reanalysis datasets, we demonstrated the effectiveness of the ConvGRU network in predicting future SST maps in the ENSO region. The ConvGRU network outperformed existing models in various scenarios, including the Niño 3.4 index prediction, showcasing its capabilities in downstream applications. We also evaluated the performance of the network in predicting other climate-related tasks, such as predicting monthly air temperature over a large portion of the global surface, which further demonstrate its potential for accurate spatio-temporal sequence predictions.

In Chapter 4, we introduces two actively adaptive algorithms for piecewise-stationary cascading bandit, namely `GLRT-CascadeUCB` and `GLRT-CascadeKL-UCB`. It is analytically established that `GLRT-CascadeUCB` and `GLRT-CascadeKL-UCB` achieve the same nearly optimal regret upper bound on the order of $\mathcal{O}\left(\sqrt{NLT\log T}\right)$, which matches our minimax regret lower bound up to a $\sqrt{\log T}$ factor. Compared with existing algorithms that adopt passively adaptive approach such as CascadeSWUCB and CascadeDUCB, our new regret upper bounds are reduced by $\mathcal{O}(\sqrt{L})$ and $\mathcal{O}(\sqrt{L \log T})$ respectively. Numerical tests on both synthetic and real-world data show the improved efficiency of the proposed algorithms.

Chapter 5 introduces a new MAB formulation – adversarial graphical contextual bandits – which leverage both contexts and side observations. Two efficient algorithms, EXP3-LGC-U and EXP3-LGC-IX, are proposed, with EXP3-LGC-IX for a special class of problems and EXP3-LGC-U for more general cases. Under mild assumptions, it is analytically demonstrated that the proposed algorithms achieve the regret $\widetilde{\mathcal{O}}(\sqrt{\alpha(G)dT})$ for both directed and undirected graph settings.

In Chapter 6, we study the joint community detection and phase synchronization problem from an MLE perspective and provide the new insight that its MLE formulation has a *multi-frequency* nature. We then propose two methods, the spectral method based on the novel MF-CPQR factorization and the iterative MF-GPM, to tackle the MLE formulation of the joint estimation problem, where the latter one requires the initialization from spectral methods. Numerical experiments demonstrate the advantage of our proposed algorithms against existing algorithms.

Almost tune-free SVRG and SARAH were developed in Chapter 7. Besides the BB step size for eliminating the tuning for step size, the key insights are that both i) averaging, as well as ii) the number of inner loop iterations should be adjusted according to the BB step size. Specific major findings include: i) estimate sequence based provably linear convergence of SVRG and SARAH, which enabled new types of averaging for efficient variance reduction; ii) theoretical guarantees of BB-SVRG and BB-SARAH with different types of averaging; and iii) implementable tune-free variance reduction algorithms. The efficacy of the tune-free BB-SVRG and BB-SARAH were corroborated numerically.

A parameter-free FW variant, ExtraFW, is introduced and analyzed in Chapter 8. ExtraFW leverages two gradient evaluations per iteration to update in a PC manner. We show that ExtraFW converges at $\mathcal{O}(\frac{1}{k})$ on general problems, while achieving a faster rate $\mathcal{O}(\frac{TLD^2}{k^2})$ on certain types of constraint sets including active ℓ_1, ℓ_2, and n-support norm balls. Given the possibility of acceleration, ExtraFW is thus a competitive alternative to FW. The efficiency of ExtraFW is validated on tasks such as i) binary classification with different constraints, where ExtraFW can be even faster than NAG, and ii) matrix completion where ExtraFW finds solutions with lower optimality error and rank rapidly.

REFERENCES

[1] Y. Fan, Y. Khoo, and Z. Zhao, "A spectral method for joint community detection and orthogonal group synchronization," *SIAM Journal on Matrix Analysis and Applications*, vol. 44, no. 2, pp. 781–821, 2023.

[2] S. Chen, X. Cheng, and A. M.-C. So, "Non-convex joint community detection and group synchronization via generalized power method," *arXiv preprint arXiv:2112.14204*, 2021.

[3] T. M. Mitchell, *Machine Learning.* McGraw-Hill, 1997, vol. 1, no. 9.

[4] L. Wang, L. Wang, M. Georgieva, P. Machado, A. Ulagappa, S. Ahmed, Y. Lu, A. Bakshi, and F. Ghassemi, "Robust nonparametric distribution forecast with backtest-based bootstrap and adaptive residual selection," in *International Conference on Acoustics, Speech and Signal Processing.* IEEE, 2022, pp. 3903–3907.

[5] L. Wang, S. Ammons, V. M. Hur, R. L. Sriver, and Z. Zhao, "Convolutional GRU network for seasonal prediction of the El Niño-Southern Oscillation," *arXiv preprint arXiv:2306.10443*, 2023.

[6] B. Kveton, C. Szepesvari, Z. Wen, and A. Ashkan, "Cascading bandits: Learning to rank in the cascade model," in *International Conference on Machine Learning.* PMLR, 2015, pp. 767–776.

[7] L. Wang, H. Zhou, B. Li, L. R. Varshney, and Z. Zhao, "Near-optimal algorithms for piecewise-stationary cascading bandits," in *International Conference on Acoustics, Speech and Signal Processing.* IEEE, 2021, pp. 3365–3369.

[8] L. Wang, B. Li, H. Zhou, G. B. Giannakis, L. R. Varshney, and Z. Zhao, "Adversarial linear contextual bandits with graph-structured side observations," in *the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, 2021, pp. 10 156–10 164.

[9] L. Wang and Z. Zhao, "Multi-frequency joint community detection and phase synchronization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 9, pp. 162–174, 2023.

[10] B. Li, L. Wang, and G. B. Giannakis, "Almost tune-free variance reduction," in *International Conference on Machine Learning.* PMLR, 2020, pp. 5969–5978.

[11] B. Li, L. Wang, G. B. Giannakis, and Z. Zhao, "Enhancing parameter-free Frank Wolfe with an extra subproblem," in *the AAAI Conference on Artificial Intelligence*, vol. 35, no. 9, 2021, pp. 8324–8331.

[12] M. J. McPhaden, S. E. Zebiak, and M. H. Glantz, "ENSO as an integrating concept in earth science," *Science*, vol. 314, no. 5806, pp. 1740–1745, 2006.

[13] K. Fraedrich and K. Müller, "Climate anomalies in Europe associated with ENSO extremes," *International Journal of Climatology*, vol. 12, no. 1, pp. 25–31, 1992.

[14] K. K. Kumar, B. Rajagopalan, and M. A. Cane, "On the weakening relationship between the Indian monsoon and ENSO," *Science*, vol. 284, no. 5423, pp. 2156–2159, 1999.

[15] M. H. Glantz et al., *Currents of change: Impacts of El Niño and La Niña on climate and society.* Cambridge University Press, 2001.

[16] A. R. Cook, L. M. Leslie, D. B. Parsons, and J. T. Schaefer, "The impact of El Niño–Southern Oscillation (ENSO) on winter and early spring US tornado outbreaks," *Journal of Applied Meteorology and Climatology*, vol. 56, no. 9, pp. 2455–2478, 2017.

[17] S. J. Camargo and A. H. Sobel, "Western North Pacific tropical cyclone intensity and ENSO," *Journal of Climate*, vol. 18, no. 15, pp. 2996–3006, 2005.

[18] D. T. Squire, D. Richardson, J. S. Risbey, A. S. Black, V. Kitsios, R. J. Matear, D. Monselesan, T. S. Moore, and C. R. Tozer, "Likelihood of unprecedented drought and fire weather during Australia's 2019 megafires," *Climate and Atmospheric Science*, vol. 4, no. 1, p. 64, 2021.

[19] V. Trouet, A. H. Taylor, A. M. Carleton, and C. N. Skinner, "Interannual variations in fire weather, fire extent, and synoptic-scale circulation patterns in northern California and Oregon," *Theoretical and Applied Climatology*, vol. 95, pp. 349–360, 2009.

[20] E. M. Rasmusson and T. H. Carpenter, "Variations in tropical sea surface temperature and surface wind fields associated with the Southern Oscillation/El Niño," *Monthly Weather Review*, vol. 110, no. 5, pp. 354–384, 1982.

[21] K. E. Trenberth, "The definition of El Niño," *Bulletin of the American Meteorological Society*, vol. 78, no. 12, pp. 2771–2778, 1997.

[22] K. E. Trenberth and D. P. Stepaniak, "Indices of El Niño evolution," *Journal of Climate*, vol. 14, no. 8, pp. 1697–1701, 2001.

[23] J. W. Hansen, A. Challinor, A. Ines, T. Wheeler, and V. Moron, "Translating climate forecasts into agricultural terms: Advances and challenges," *Climate Research*, vol. 33, no. 1, pp. 27–41, 2006.

[24] G. Hammer, J. Hansen, J. Phillips, J. Mjelde, H. Hill, A. Love, and A. Potgieter, "Advances in application of climate prediction in agriculture," *Agricultural Systems*, vol. 70, no. 2-3, pp. 515–553, 2001.

[25] A. G. Barnston, M. K. Tippett, M. Ranganathan, and M. L. L'Heureux, "Deterministic skill of ENSO predictions from the North American Multimodel Ensemble," *Climate Dynamics*, vol. 53, pp. 7215–7234, 2019.

[26] H. Ding, M. Newman, M. A. Alexander, and A. T. Wittenberg, "Skillful climate forecasts of the tropical Indo-Pacific ocean using model-analogs," *Journal of Climate*, vol. 31, no. 14, pp. 5437–5459, 2018.

[27] H. Ding, M. Newman, M. A. Alexander, and A. T. Wittenberg, "Diagnosing secular variations in retrospective ENSO seasonal forecast skill using CMIP5 model-analogs," *Geophysical Research Letters*, vol. 46, no. 3, pp. 1721–1730, 2019.

[28] X. Wang, J. Slawinska, and D. Giannakis, "Extended-range statistical ENSO prediction through operator-theoretic techniques for nonlinear dynamics," *Scientific Reports*, vol. 10, no. 1, pp. 1–15, 2020.

[29] C. Penland and T. Magorian, "Prediction of Niño 3 sea surface temperatures using linear inverse modeling," *Journal of Climate*, vol. 6, no. 6, pp. 1067–1076, 1993.

[30] C. Penland and P. D. Sardeshmukh, "The optimal growth of tropical sea surface temperature anomalies," *Journal of Climate*, vol. 8, no. 8, pp. 1999–2024, 1995.

[31] D. Chapman, M. A. Cane, N. Henderson, D. E. Lee, and C. Chen, "A vector autoregressive ENSO prediction model," *Journal of Climate*, vol. 28, no. 21, pp. 8511–8520, 2015.

[32] E. N. Lorenz, "Atmospheric predictability as revealed by naturally occurring analogues," *Journal of Atmospheric Sciences*, vol. 26, no. 4, pp. 636–646, 1969.

[33] Z. Zhao and D. Giannakis, "Analog forecasting with dynamics-adapted kernels," *Nonlinearity*, vol. 29, no. 9, p. 2888, 2016.

[34] D. Burov, D. Giannakis, K. Manohar, and A. Stuart, "Kernel analog forecasting: Multiscale test problems," *Multiscale Modeling and Simulation*, vol. 19, no. 2, pp. 1011–1040, 2021.

[35] Y.-G. Ham, J.-H. Kim, and J.-J. Luo, "Deep learning for multi-year ENSO forecasts," *Nature*, vol. 573, no. 7775, pp. 568–572, 2019.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] A. Huang, B. Vega-Westhoff, and R. L. Sriver, "Analyzing El Niño–Southern Oscillation predictability using long-short-term-memory models," *Earth and Space Science*, vol. 6, no. 2, pp. 212–221, 2019.

[38] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems*, 2015.

[39] X. Shi, Z. Gao, L. Lausen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Deep learning for precipitation nowcasting: A benchmark and a new model," in *Advances in Neural Information Processing Systems*, 2017.

[40] N. Ballas, L. Yao, C. Pal, and A. Courville, "Delving deeper into convolutional networks for learning video representations," *arXiv preprint arXiv:1511.06432*, 2015.

[41] P. D. Larson, "Designing and managing the supply chain: Concepts, strategies, and case studies," *Journal of Business Logistics*, vol. 22, no. 1, pp. 259–261, 2001.

[42] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and practice*. OTexts, 2018.

[43] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[44] S. S. Rangapuram, M. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, "Deep state space models for time series forecasting," in *Advances in Neural Information Processing Systems*, 2018, pp. 7796–7805.

[45] D. H. Bailey, J. Borwein, M. Lopez de Prado, and Q. J. Zhu, “The probability of backtest overfitting,” *Journal of Computational Finance, Forthcoming*, 2016.

[46] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, “The M4 competition: Results, findings, conclusion and way forward,” *International Journal of Forecasting*, vol. 34, no. 4, pp. 802–808, 2018.

[47] A. Alexandrov, K. Benidis, M. Bohlke-Schneider, V. Flunkert, J. Gasthaus, T. Januschowski, D. C. Maddix, S. Rangapuram, D. Salinas, J. Schulz et al., “GluonTS: Probabilistic and neural time series modeling in python,” *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 4629–4634, 2020.

[48] S. Li, A. Karatzoglou, and C. Gentile, “Collaborative filtering bandits,” in *International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 2016, pp. 539–548.

[49] G. E. Dupret and B. Piwowarski, “A user browsing model to predict search engine click data from past observations.” in *International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 2008, pp. 331–338.

[50] M. Zoghi, T. Tunys, M. Ghavamzadeh, B. Kveton, C. Szepesvari, and Z. Wen, “Online learning to rank in stochastic click models,” in *International Conference on Machine Learning.* PMLR, 2017, pp. 4199–4208.

[51] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey, “An experimental comparison of click position-bias models,” in *International Conference on Web Search and Data Mining.* ACM, 2008, pp. 87–94.

[52] W. C. Cheung, V. Tan, and Z. Zhong, “A Thompson sampling algorithm for cascading bandits,” in *International Conference on Artificial Intelligence and Statistics.* PMLR, 2019, pp. 438–447.

[53] R. Jagerman, I. Markov, and M. de Rijke, “When people change their mind: Off-policy evaluation in non-stationary recommendation environments,” in *International Conference on Web Search and Data Mining.* ACM, 2019, pp. 447–455.

[54] J. Y. Yu and S. Mannor, “Piecewise-stationary bandit problems with side observations,” in *International Conference on Machine Learning.* PMLR, 2009, pp. 1177–1184.

[55] F. S. Pereira, J. Gama, S. de Amo, and G. M. Oliveira, “On analyzing user preference dynamics with temporal social networks,” *Machine Learning*, vol. 107, no. 11, pp. 1745–1773, 2018.

[56] C. Li and M. de Rijke, "Cascading non-stationary bandits: Online learning to rank in the non-stationary cascade model," in *International Joint Conference on Artificial Intelligence*, 2019, pp. 2859–2865.

[57] A. Garivier and E. Moulines, "On upper-confidence bound policies for switching bandit problems," in *International Conference on Algorithmic Learning Theory.* Springer, 2011, pp. 174–188.

[58] O. Besbes, Y. Gur, and A. Zeevi, "Stochastic multi-armed-bandit problem with non-stationary rewards," in *Advances in Neural Information Processing Systems*, 2014, pp. 199–207.

[59] L. Wei and V. Srivatsva, "On abruptly-changing and slowly-varying multiarmed bandit problems," in *American Control Conference.* IEEE, 2018, pp. 6291–6296.

[60] Y. Cao, Z. Wen, B. Kveton, and Y. Xie, "Nearly optimal adaptive procedure with change detection for piecewise-stationary bandit," in *International Conference on Artificial Intelligence and Statistics.* PMLR, 2019, pp. 418–427.

[61] F. Liu, J. Lee, and N. Shroff, "A change-detection based framework for piecewise-stationary multi-armed bandit problem," in *the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[62] L. Besson and E. Kaufmann, "The generalized likelihood ratio test meets KL-UCB: An improved algorithm for piece-wise non-stationary bandits," *arXiv preprint arXiv:1902.01575*, 2019.

[63] P. Auer, P. Gajane, and R. Ortner, "Adaptively tracking the best bandit arm with an unknown number of distribution changes," in *Conference on Learning Theory.* PMLR, 2019, pp. 138–158.

[64] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954.

[65] D. V. Hinkley, "Inference about the change-point from cumulative sum tests," *Biometrika*, vol. 58, no. 3, pp. 509–523, 1971.

[66] A. Willsky and H. Jones, "A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems," *IEEE Transactions on Automatic Control*, vol. 21, no. 1, pp. 108–112, 1976.

[67] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *International Conference on World Wide Web.* ACM, 2010, pp. 661–670.

[68] W. Chu, L. Li, L. Reyzin, and R. Schapire, "Contextual bandits with linear payoff functions," in *International Conference on Artificial Intelligence and Statistics.* PMLR, 2011, pp. 208–214.

[69] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," in *Advances in Neural Information Processing Systems*, 2011, pp. 2312–2320.

[70] A. Agarwal, D. Hsu, S. Kale, J. Langford, L. Li, and R. Schapire, "Taming the monster: A fast and simple algorithm for contextual bandits," in *International Conference on Machine Learning.* PMLR, 2014, pp. 1638–1646.

[71] S. Mannor and O. Shamir, "From bandits to experts: On the value of side-observations," in *Advances in Neural Information Processing Systems*, 2011, pp. 684–692.

[72] N. Alon, N. Cesa-Bianchi, C. Gentile, and Y. Mansour, "From bandits to experts: A tale of domination and independence," in *Advances in Neural Information Processing Systems*, 2013, pp. 1610–1618.

[73] N. Alon, N. Cesa-Bianchi, O. Dekel, and T. Koren, "Online learning with feedback graphs: Beyond bandits," in *Conference on Learning Theory.* PMLR, 2015, pp. 1–13.

[74] N. Alon, N. Cesa-Bianchi, C. Gentile, S. Mannor, Y. Mansour, and O. Shamir, "Nonstochastic multi-armed bandits with graph-structured feedback," *SIAM Journal on Computing*, vol. 46, no. 6, pp. 1785–1826, 2017.

[75] A. Tewari and S. A. Murphy, "From ads to interventions: Contextual bandits in mobile health," in *Mobile Health.* Springer, 2017, pp. 495–517.

[76] B. Li, T. Chen, and G. B. Giannakis, "Bandit online learning with unknown delays," in *International Conference on Artificial Intelligence and Statistics.* PMLR, 2019, pp. 993–1002.

[77] H. Zhou, L. Wang, L. Varshney, and E.-P. Lim, "A near-optimal change-detection based algorithm for piecewise-stationary combinatorial semi-bandits," in *the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 6933–6940.

[78] F. Liu, Z. Zheng, and N. Shroff, "Analysis of Thompson sampling for graphical bandits without the graphs," in *Conference on Uncertainty in Artificial Intelligence.* AUAI, 2018, pp. 13–22.

[79] I. Lobel, E. Sadler, and L. R. Varshney, "Customer referral incentives and social media," *Management Science*, vol. 63, no. 10, pp. 3514–3529, 2017.

[80] S. Agrawal and N. Goyal, "Analysis of Thompson sampling for the multi-armed bandit problem," in *Conference on Learning Theory*. PMLR, 2012, pp. 1–26.

[81] G. Neu and J. Olkhovskaya, "Efficient and robust algorithms for adversarial linear contextual bandits," in *Conference on Learning Theory*. PMLR, 2020, pp. 1–20.

[82] T. Kocák, G. Neu, M. Valko, and R. Munos, "Efficient learning by implicit exploration in bandit problems with side observations," in *Advances in Neural Information Processing Systems*, 2014, pp. 613–621.

[83] E. Abbe, "Community detection and stochastic block models: Recent developments," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6446–6531, 2017.

[84] A. Singer, "Angular synchronization by eigenvectors and semidefinite programming," *Applied and Computational Harmonic Analysis*, vol. 30, no. 1, pp. 20–36, 2011.

[85] Z. Chen, L. Li, and J. Bruna, "Supervised community detection with line graph neural networks," in *International Conference on Learning Representations*, 2019.

[86] K. Berahmand, M. Mohammadi, A. Faroughi, and R. P. Mohammadiani, "A novel method of spectral clustering in attributed networks by constructing parameter-free affinity matrix," *Cluster Computing*, vol. 25, no. 2, pp. 869–888, 2022.

[87] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[88] K. Berahmand, A. Bouyer, and M. Vasighi, "Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 4, pp. 1021–1033, 2018.

[89] A. Singer, Z. Zhao, Y. Shkolnisky, and R. Hadani, "Viewing angle classification of cryo-electron microscopy images using eigenvectors," *SIAM Journal on Imaging Sciences*, vol. 4, no. 2, pp. 723–759, 2011.

[90] Z. Zhao and A. Singer, "Rotationally invariant image representation for viewing direction classification in cryo-EM," *Journal of Structural Biology*, vol. 186, no. 1, pp. 153–166, 2014.

[91] M. Zamiri, T. Bahraini, and H. S. Yazdi, "MVDF-RSC: Multi-view data fusion via robust spectral clustering for geo-tagged image tagging," *Expert Systems with Applications*, vol. 173, p. 114657, 2021.

[92] E. Abbe, A. S. Bandeira, and G. Hall, "Exact recovery in the stochastic block model," *IEEE Transactions on Information Theory*, vol. 62, no. 1, pp. 471–487, 2015.

[93] E. Abbe and C. Sandon, "Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery," in *Annual Symposium on Foundations of Computer Science*. IEEE, 2015, pp. 670–688.

[94] E. Abbe, J. Fan, K. Wang, and Y. Zhong, "Entrywise eigenvector analysis of random matrices with low expected rank," *The Annals of Statistics*, vol. 48, no. 3, pp. 1452–1474, 2020.

[95] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová, and P. Zhang, "Spectral redemption in clustering sparse networks," *Proceedings of the National Academy of Sciences*, vol. 110, no. 52, pp. 20 935–20 940, 2013.

[96] L. Massoulié, "Community detection thresholds and the weak Ramanujan property," in *ACM Symposium on Theory of Computing*, 2014, pp. 694–703.

[97] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, vol. 14, 2001.

[98] V. Vu, "A simple SVD algorithm for finding hidden partitions," *Combinatorics, Probability and Computing*, vol. 27, no. 1, pp. 124–140, 2018.

[99] S.-Y. Yun and A. Proutiere, "Accurate community detection in the stochastic block model via spectral algorithms," *arXiv preprint arXiv:1412.7335*, 2014.

[100] L. Su, W. Wang, and Y. Zhang, "Strong consistency of spectral clustering for stochastic block models," *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 324–338, 2019.

[101] F. McSherry, "Spectral partitioning of random graphs," in *IEEE Symposium on Foundations of Computer Science*. IEEE, 2001, pp. 529–537.

[102] A. A. Amini and E. Levina, "On semidefinite relaxations for the block model," *The Annals of Statistics*, vol. 46, no. 1, pp. 149–179, 2018.

[103] A. S. Bandeira, “Random Laplacian matrices and convex relaxations,” *Foundations of Computational Mathematics*, vol. 18, no. 2, pp. 345–379, 2018.

[104] O. Guédon and R. Vershynin, “Community detection in sparse networks via Grothendieck’s inequality,” *Probability Theory and Related Fields*, vol. 165, no. 3, pp. 1025–1049, 2016.

[105] B. Hajek, Y. Wu, and J. Xu, “Achieving exact cluster recovery threshold via semidefinite programming,” *IEEE Transactions on Information Theory*, vol. 62, no. 5, pp. 2788–2797, 2016.

[106] B. Hajek, Y. Wu, and J. Xu, “Achieving exact cluster recovery threshold via semidefinite programming: Extensions,” *IEEE Transactions on Information Theory*, vol. 62, no. 10, pp. 5918–5937, 2016.

[107] A. Perry and A. S. Wein, “A semidefinite program for unbalanced multisection in the stochastic block model,” in *International Conference on Sampling Theory and Applications.* IEEE, 2017, pp. 64–67.

[108] Y. Fei and Y. Chen, “Exponential error rates of SDP for block models: Beyond Grothendieck’s inequality,” *IEEE Transactions on Information Theory*, vol. 65, no. 1, pp. 551–571, 2018.

[109] X. Li, Y. Chen, and J. Xu, “Convex relaxation methods for community detection,” *Statistical Science*, vol. 36, no. 1, pp. 2–15, 2021.

[110] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, “Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications,” *Physical Review E*, vol. 84, no. 6, p. 066106, 2011.

[111] Y. Chen and A. J. Goldsmith, “Information recovery from pairwise measurements,” in *IEEE International Symposium on Information Theory.* IEEE, 2014, pp. 2012–2016.

[112] S. Zhang and Y. Huang, “Complex quadratic optimization and semidefinite programming,” *SIAM Journal on Optimization*, vol. 16, no. 3, pp. 871–890, 2006.

[113] M. Cucuringu, A. Singer, and D. Cowburn, “Eigenvector synchronization, graph rigidity and the molecule problem,” *Information and Inference: A Journal of the IMA*, vol. 1, no. 1, pp. 21–67, 2012.

[114] K. N. Chaudhury, Y. Khoo, and A. Singer, “Global registration of multiple point clouds using semidefinite programming,” *SIAM Journal on Optimization*, vol. 25, no. 1, pp. 468–501, 2015.

[115] A. S. Bandeira, C. Kennedy, and A. Singer, “Approximating the little Grothendieck problem over the orthogonal and unitary groups,” *Mathematical Programming*, vol. 160, no. 1, pp. 433–475, 2016.

[116] A. S. Bandeira, N. Boumal, and A. Singer, “Tightness of the maximum likelihood semidefinite relaxation for angular synchronization,” *Mathematical Programming*, vol. 163, no. 1, pp. 145–167, 2017.

[117] N. Boumal, “Nonconvex phase synchronization,” *SIAM Journal on Optimization*, vol. 26, no. 4, pp. 2355–2377, 2016.

[118] H. Liu, M.-C. Yue, and A. Man-Cho So, “On the estimation performance and convergence rate of the generalized power method for phase synchronization,” *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2426–2446, 2017.

[119] Y. Zhong and N. Boumal, “Near-optimal bounds for phase synchronization,” *SIAM Journal on Optimization*, vol. 28, no. 2, pp. 989–1016, 2018.

[120] A. S. Bandeira, Y. Chen, R. R. Lederman, and A. Singer, “Non-unique games over compact groups and orientation estimation in cryo-EM,” *Inverse Problems*, vol. 36, no. 6, pp. 1–39, 2020.

[121] A. Perry, A. S. Wein, A. S. Bandeira, and A. Moitra, “Message-passing algorithms for synchronization problems over compact groups,” *Communications on Pure and Applied Mathematics*, vol. 71, no. 11, pp. 2275–2322, 2018.

[122] T. Gao and Z. Zhao, “Multi-frequency phase synchronization,” in *International Conference on Machine Learning.* PMLR, 2019, pp. 2132–2141.

[123] Y. Fan, Y. Khoo, and Z. Zhao, “Joint community detection and rotational synchronization via semidefinite programming,” *SIAM Journal on Mathematics of Data Science*, vol. 4, no. 3, pp. 1052–1081, 2022.

[124] J. Frank, *Three-dimensional electron microscopy of macromolecular assemblies: Visualization of biological molecules in their native state.* Oxford University Press, 2006.

[125] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course.* Springer Science and Business Media, 2004, vol. 87.

[126] H. Robbins and S. Monro, “A stochastic approximation method,” *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

[127] L. Bottou, F. E. Curtis, and J. Nocedal, “Optimization methods for large-scale machine learning,” *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018.

[128] R. Johnson and T. Zhang, “Accelerating stochastic gradient descent using predictive variance reduction,” in *Advances in Neural Information Processing Systems*, 2013, pp. 315–323.

[129] N. L. Roux, M. Schmidt, and F. R. Bach, “A stochastic gradient method with an exponential convergence rate for finite training sets,” in *Advances in Neural Information Processing Systems*, 2012, pp. 2663–2671.

[130] A. Defazio, F. Bach, and S. Lacoste-Julien, “SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives,” in *Advances in Neural Information Processing Systems*, 2014, pp. 1646–1654.

[131] J. Mairal, “Optimization with first-order surrogate functions,” in *International Conference on Machine Learning*. PMLR, 2013, pp. 783–791.

[132] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč, “SARAH: A novel method for machine learning problems using stochastic recursive gradient,” in *International conference on machine learning*. PMLR, 2017, pp. 2613–2621.

[133] J. Konečnỳ and P. Richtárik, “Semi-stochastic gradient descent methods,” *Frontiers in Applied Mathematics and Statistics*, vol. 3, p. 9, 2017.

[134] L. Lei, C. Ju, J. Chen, and M. I. Jordan, “Non-convex finite-sum optimization via SCSG methods,” in *Advances in Neural Information Processing Systems*. PMLR, 2017, pp. 2348–2358.

[135] D. Kovalev, S. Horváth, and P. Richtárik, “Don’t jump through hoops and remove those loops: SVRG and Katyusha are better without the outer loop,” in *Algorithmic Learning Theory*. PMLR, 2020, pp. 451–467.

[136] B. Li, M. Ma, and G. B. Giannakis, “On the convergence of SARAH and beyond,” in *International Conference on Artificial Intelligence and Statistics*. PRML, 2020.

[137] S. Shalev-Shwartz and T. Zhang, “Stochastic dual coordinate ascent methods for regularized loss minimization,” *The Journal of Machine Learning Research*, vol. vol. 14, pp. pp. 567–599, Feb. 2013.

[138] A. Agarwal and L. Bottou, “A lower bound for the optimization of finite sums,” in *International Conference on Machine Learning.* PRML, 2015, pp. 78–86.

[139] J. Barzilai and J. M. Borwein, “Two-point step size gradient methods,” *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[140] C. Tan, S. Ma, Y.-H. Dai, and Y. Qian, “Barzilai-Borwein step size for stochastic gradient descent,” in *Advances in Neural Information Processing Systems*, 2016, pp. 685–693.

[141] Z. Yang, Z. Chen, and C. Wang, “Accelerating mini-batch SARAH by step size rules,” *Information Sciences*, vol. 558, pp. 157–173, 2021.

[142] R. M. Freund, P. Grigas, and R. Mazumder, “An extended Frank–Wolfe method with “in-face” directions, and its application to low-rank matrix completion,” *SIAM Journal on Optimization*, vol. 27, no. 1, pp. 319–346, 2017.

[143] Z. Harchaoui, A. Juditsky, and A. Nemirovski, “Conditional gradient algorithms for norm-regularized smooth convex optimization,” *Mathematical Programming*, vol. 152, no. 1-2, pp. 75–112, 2015.

[144] Z. Allen-Zhu and L. Orecchia, “Linear coupling: An ultimate unification of gradient and mirror descent,” *arXiv preprint arXiv:1407.1537*, 2014.

[145] Y. Nesterov, “Universal gradient methods for convex optimization problems,” *Mathematical Programming*, vol. 152, no. 1-2, pp. 381–404, 2015.

[146] B. Li, A. Sadeghi, and G. Giannakis, “Heavy ball momentum for conditional gradient,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 244–21 255, 2021.

[147] M. Frank and P. Wolfe, “An algorithm for quadratic programming,” *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.

[148] M. Jaggi, “Revisiting Frank-Wolfe: Projection-free sparse convex optimization.” in *International Conference on Machine Learning.* PRML, 2013, pp. 427–435.

[149] S. Lacoste-Julien and M. Jaggi, “On the global linear convergence of Frank-Wolfe optimization variants,” in *Advances in Neural Information Processing Systems*, 2015, pp. 496–504.

[150] D. Garber and E. Hazan, “Faster rates for the Frank-Wolfe method over strongly-convex sets,” in *International Conference on Machine Learning.* PMLR, 2015.

[151] S. Lacoste-Julien, M. Jaggi, M. W. Schmidt, and P. Pletscher, "Block-coordinate Frank-Wolfe optimization for structural SVMs," in *International Conference on Machine Learning.* PMLR, 2013, pp. 53–61.

[152] A. Joulin, K. Tang, and L. Fei-Fei, "Efficient image and video co-localization with Frank-Wolfe algorithm," in *European Conference on Computer Vision.* Springer, 2014, pp. 253–268.

[153] G. Luise, S. Salzo, M. Pontil, and C. Ciliberto, "Sinkhorn barycenters with free support via Frank-Wolfe algorithm," in *Advances in Neural Information Processing Systems*, 2019, pp. 9318–9329.

[154] A. Mokhtari, H. Hassani, and A. Karbasi, "Stochastic conditional gradient methods: From convex minimization to submodular maximization," *The Journal of Machine Learning Research*, 2020.

[155] G. Lan, "The complexity of large-scale convex programming under a linear optimization oracle," *arXiv preprint arXiv:1309.5550*, 2013.

[156] J. Guélat and P. Marcotte, "Some comments on Wolfe's 'away step'," *Mathematical Programming*, vol. 35, no. 1, pp. 110–119, 1986.

[157] F. Pedregosa, A. Askari, G. Negiar, and M. Jaggi, "Step-size adaptivity in projection-free optimization," *arXiv preprint arXiv:1806.05123*, 2018.

[158] G. Braun, S. Pokutta, D. Tu, and S. Wright, "Blended conditonal gradients," in *International Conference on Machine Learning.* PMLR, 2019, pp. 735–743.

[159] D. Garber and O. Meshi, "Linear-memory and decomposition-invariant linearly convergent conditional gradient algorithm for structured polytopes," in *Advances in Neural Information Processing Systems*, 2016, pp. 1001–1009.

[160] E. S. Levitin and B. T. Polyak, "Constrained minimization methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 6, no. 5, pp. 1–50, 1966.

[161] T. Kerdreux, A. d'Aspremont, and S. Pokutta, "Projection-free optimization on uniformly convex sets," in *International Conference on Artificial Intelligence and Statistics.* PMLR, 2021, pp. 19–27.

[162] F. Bach, "On the effectiveness of Richardson extrapolation in data science," *SIAM Journal on Mathematics of Data Science*, vol. 3, no. 4, pp. 1251–1277, 2021.

[163] B. Li, M. Coutino, G. B. Giannakis, and G. Leus, "How does momentum help Frank Wolfe?" *arXiv preprint arXiv:1908.09345*, 2020.

[164] G. Korpelevich, "The extragradient method for finding saddle points and other problems," *Matecon: Translations of Russian and East European Mathematical Economics*, vol. 12, pp. 747–756, 1976.

[165] A. Nemirovski, "Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems," *SIAM Journal on Optimization*, vol. 15, no. 1, pp. 229–251, 2004.

[166] J. Diakonikolas and L. Orecchia, "Accelerated extra-gradient descent: A novel accelerated first-order method," *arXiv preprint arXiv:1706.04680*, 2017.

[167] A. Kavis, K. Y. Levy, F. Bach, and V. Cevher, "Unixgrad: A universal, adaptive algorithm with optimal guarantees for constrained optimization," in *Advances in Neural Information Processing Systems*, 2019, pp. 6257–6266.

[168] L. Wang and Z. Zhao, "Two-dimensional tomography from noisy projection tilt series taken at unknown view angles with non-uniform distribution," in *International Conference on Image Processing.* IEEE, 2019, pp. 1242–1246.

[169] Y. Zhang, J. M. Wallace, and D. S. Battisti, "ENSO-like interdecadal variability: 1900–93," *Journal of Climate*, vol. 10, no. 5, pp. 1004–1020, 1997.

[170] J. C. Hess, C. A. Scott, G. L. Hufford, and M. D. Fleming, "El Niño and its impact on fire weather conditions in Alaska," *International Journal of Wildland Fire*, vol. 10, no. 1, pp. 1–13, 2001.

[171] L. R. Medsker and L. Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, pp. 64–67, 2001.

[172] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*, 2014, pp. 103–111.

[173] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[174] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," in *Advances in Neural Information Processing Systems*, 2016.

[175] Y. Wang, H. Wu, J. Zhang, Z. Gao, J. Wang, S. Y. Philip, and M. Long, "Predrnn: A recurrent neural network for spatiotemporal predictive learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 2208–2225, 2022.

[176] A. H. d. O. Fonseca, E. Zappala, J. O. Caro, and D. van Dijk, "Continuous spatiotemporal transformers," *arXiv preprint arXiv:2301.13338*, 2023.

[177] Z. Yang, X. Yang, and Q. Lin, "TCTN: A 3D-temporal convolutional transformer network for spatiotemporal predictive learning," *arXiv preprint arXiv:2112.01085*, 2021.

[178] R. Keisler, "Forecasting global weather with graph neural networks," *arXiv preprint arXiv:2202.07575*, 2022.

[179] R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, A. Pritzel, S. Ravuri, T. Ewalds, F. Alet, Z. Eaton-Rosen et al., "Graphcast: Learning skillful medium-range global weather forecasting," *arXiv preprint arXiv:2212.12794*, 2022.

[180] S. Ravuri, K. Lenc, M. Willson, D. Kangin, R. Lam, P. Mirowski, M. Fitzsimons, M. Athanassiadou, S. Kashem, S. Madge et al., "Skilful precipitation nowcasting using deep generative models of radar," *Nature*, vol. 597, no. 7878, pp. 672–677, 2021.

[181] T. Eisner, B. Farkas, M. Haase, and R. Nagel, *Operator Theoretic Aspects of Ergodic Theory.* Springer, 2015, vol. 272.

[182] D. Giannakis, "Data-driven spectral decomposition and forecasting of ergodic dynamical systems," *Applied and Computational Harmonic Analysis*, vol. 47, no. 2, pp. 338–396, 2019.

[183] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[184] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[185] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization." *The Journal of Machine Learning Research*, vol. 12, no. 7, 2011.

[186] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga et al., "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019.

[187] P. R. Gent, G. Danabasoglu, L. J. Donner, M. M. Holland, E. C. Hunke, S. R. Jayne, D. M. Lawrence, R. B. Neale, P. J. Rasch, M. Vertenstein et al., "The community climate system model version 4," *Journal of Climate*, vol. 24, no. 19, pp. 4973–4991, 2011.

[188] T. L. Delworth, W. F. Cooke, A. Adcroft, M. Bushuk, J.-H. Chen, K. A. Dunne, P. Ginoux, R. Gudgel, R. W. Hallberg, L. Harris et al., "SPEAR: The next generation GFDL modeling system for seasonal to multidecadal prediction and projection," *Journal of Advances in Modeling Earth Systems*, vol. 12, no. 3, 2020.

[189] D. P. Van Vuuren, E. Kriegler, B. C. O'Neill, K. L. Ebi, K. Riahi, T. R. Carter, J. Edmonds, S. Hallegatte, T. Kram, R. Mathur et al., "A new scenario framework for climate change research: Scenario matrix architecture," *Climatic C*, vol. 122, no. 3, pp. 373–386, 2014.

[190] B. C. O'Neill, C. Tebaldi, D. P. Van Vuuren, V. Eyring, P. Friedlingstein, G. Hurtt, R. Knutti, E. Kriegler, J.-F. Lamarque, J. Lowe et al., "The scenario model intercomparison project (ScenarioMIP) for CMIP6," *Geoscientific Model Development*, vol. 9, no. 9, pp. 3461–3482, 2016.

[191] G. P. Compo, J. S. Whitaker, P. D. Sardeshmukh, N. Matsui, R. J. Allan, X. Yin, B. E. Gleason, R. S. Vose, G. Rutledge, P. Bessemoulin et al., "The twentieth century reanalysis project," *Quarterly Journal of the Royal Meteorological Society*, vol. 137, no. 654, pp. 1–28, 2011.

[192] O. Krueger, F. Schenk, F. Feser, and R. Weisse, "Inconsistencies between long-term trends in storminess derived from the 20CR reanalysis and observations," *Journal of Climate*, vol. 26, no. 3, pp. 868–874, 2013.

[193] G. J. Székely and M. L. Rizzo, "Brownian distance covariance," *The Annals of Applied Statistics*, pp. 1236–1265, 2009.

[194] K. AN, "Sulla determinazione empirica di una legge didistribuzione," *Giorn Dell'inst Ital Degli Att*, vol. 4, pp. 89–91, 1933.

[195] B. Lim and S. Zohren, "Time-series forecasting with deep learning: A survey," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20200209, 2021.

[196] D. N. Politis, "Model-free model-fitting and predictive distributions," *Test*, vol. 22, no. 2, pp. 183–221, 2013.

[197] L. Pan and D. N. Politis, "Bootstrap prediction intervals for linear, nonlinear and nonparametric autoregressions," *Journal of Statistical Planning and Inference*, vol. 177, pp. 1–27, 2016.

[198] J. Berkowitz and L. Kilian, "Recent developments in bootstrapping time series," *Econometric Reviews*, vol. 19, no. 1, pp. 1–48, 2000.

[199] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[200] Y. Wang, A. Smola, D. Maddix, J. Gasthaus, D. Foster, and T. Januschowski, "Deep factors for forecasting," in *International Conference on Machine Learning.* PMLR, 2019, pp. 6607–6617.

[201] R. Wen, K. Torkkola, B. Narayanaswamy, and D. Madeka, "A multi-horizon quantile recurrent forecaster," *arXiv preprint arXiv:1711.11053*, 2017.

[202] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.

[203] L. Kocsis and C. Szepesvári, "Discounted UCB," in *PASCAL Challenges Workshop*, vol. 2, 2006, pp. 51–134.

[204] O. Hadjiliadis and V. Moustakides, "Optimal and asymptotically optimal CUSUM rules for change point detection in the Brownian motion model with multiple alternatives," *Theory of Probability and Its Applications*, vol. 50, no. 1, pp. 75–85, 2006.

[205] D. Siegmund, *Sequential Analysis: Tests and Confidence Intervals.* Springer Science and Business Media, 2013.

[206] V. Draglia, A. G. Tartakovsky, and V. V. Veeravalli, "Multihypothesis sequential probability ratio tests. I. Asymptotic optimality," *IEEE Transactions on Information Theory*, vol. 45, no. 7, pp. 2448–2461, 1999.

[207] D. Siegmund and E. Venkatraman, "Using the generalized likelihood ratio statistic for sequential detection of a change-point," *The Annals of Statistics*, pp. 255–271, 1995.

[208] G. Lorden, "Procedures for reacting to a change in distribution," *The Annals of Mathematical Statistics*, pp. 1897–1908, 1971.

[209] G. V. Moustakides, "Optimal stopping times for detecting changes in distributions," *The Annals of Statistics*, vol. 14, no. 4, pp. 1379–1387, 1986.

[210] E. Kaufmann and W. M. Koolen, “Mixture Martingales revisited with applications to sequential tests and confidence intervals,” *The Journal of Machine Learning Research*, vol. 22, pp. 246–1, 2021.

[211] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time analysis of the multiarmed bandit problem,” *Machine Learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[212] O. Cappé, A. Garivier, O.-A. Maillard, R. Munos, G. Stoltz et al., “Kullback–Leibler upper confidence bounds for optimal sequential allocation,” *The Annals of Statistics*, vol. 41, no. 3, pp. 1516–1541, 2013.

[213] R. Combes, M. S. T. M. Shahi, A. Proutiere et al., “Combinatorial bandits revisited,” in *Advances in Neural Information Processing Systems*, 2015, pp. 2116–2124.

[214] N. Cesa-Bianchi and G. Lugosi, “Combinatorial bandits,” *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1404–1422, 2012.

[215] Q. Wang and W. Chen, “Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications,” in *Advances in Neural Information Processing Systems*, 2017, pp. 1161–1171.

[216] L. Li, W. Chu, J. Langford, and X. Wang, “Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms,” in *International Conference on Web Search and Data Mining*. ACM, 2011, pp. 297–306.

[217] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, “The nonstochastic multiarmed bandit problem,” *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2002.

[218] A. Cohen, T. Hazan, and T. Koren, “Online learning with feedback graphs without the graphs,” in *International Conference on Machine Learning*. PMLR, 2016, pp. 811–819.

[219] A. Rakhlin and K. Sridharan, “BISTRO: An efficient relaxation-based method for contextual bandits.” in *International Conference on Machine Learning*. PMLR, 2016, pp. 1977–1985.

[220] V. Syrgkanis, A. Krishnamurthy, and R. Schapire, “Efficient algorithms for adversarial contextual learning,” in *International Conference on Machine Learning*. PMLR, 2016, pp. 2159–2168.

[221] V. Syrgkanis, H. Luo, A. Krishnamurthy, and R. E. Schapire, “Improved regret bounds for oracle-based adversarial contextual bandits,” in *Advances in Neural Information Processing Systems*, 2016, pp. 3135–3143.

[222] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[223] T. Lattimore and C. Szepesvári, *Bandit Algorithms.* Cambridge University Press, 2020.

[224] P. Auer, N. Cesa-Bianchi, and C. Gentile, "Adaptive and self-confident on-line learning algorithms," *Journal of Computer and System Sciences*, vol. 64, no. 1, pp. 48–75, 2002.

[225] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra.* SIAM, 1997, vol. 50.

[226] R. Bulirsch, J. Stoer, and J. Stoer, *Introduction to Numerical Analysis.* Springer, 1991.

[227] P. Businger and G. Golub, "Linear least squares solutions by Householder transformations," in *Handbook for Automatic Computation.* Springer, 1971, pp. 111–118.

[228] P. Wang, H. Liu, Z. Zhou, and A. M.-C. So, "Optimal non-convex exact recovery in stochastic block model via projected power method," in *International Conference on Machine Learning.* PMLR, 2021, pp. 10 828–10 838.

[229] T. Tokuyama and J. Nakano, "Geometric algorithms for the minimum cost assignment problem," *Random Structures and Algorithms*, vol. 6, no. 4, pp. 393–406, 1995.

[230] K. Numata and T. Tokuyama, "Splitting a configuration in a simplex," *Algorithmica*, vol. 9, no. 6, pp. 649–668, 1993.

[231] A. Damle, V. Minden, and L. Ying, "Simple, direct and efficient multi-way spectral clustering," *Information and Inference: A Journal of the IMA*, vol. 8, no. 1, pp. 181–203, 2019.

[232] G. H. Golub and C. F. Van Loan, *Matrix Computations.* The Johns Hopkins University Press, 1996.

[233] G. W. Stewart, "A Krylov–Schur algorithm for large eigenproblems," *SIAM Journal on Matrix Analysis and Applications*, vol. 23, no. 3, pp. 601–614, 2002.

[234] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Statistical properties of community structure in large social and information networks," in *International Conference on World Wide Web.* ACM, 2008, pp. 695–704.

[235] L. Xiao and T. Zhang, “A proximal stochastic gradient method with progressive variance reduction,” *SIAM Journal on Optimization*, vol. 24, no. 4, pp. 2057–2075, 2014.

[236] A. Kulunchakov and J. Mairal, “Estimate sequences for variance-reduced stochastic composite optimization,” in *International Conference on Machine Learning.* PMLR, 2019, pp. 3541–3550.

[237] B. Li and G. B. Giannakis, “Enhancing sharpness-aware optimization through variance suppression,” *arXiv preprint arXiv:2309.15639*, 2023.

[238] B. Hu, S. Wright, and L. Lessard, “Dissipativity theory for accelerating stochastic variance reduction: A unified analysis of SVRG and Katyusha using semidefinite programs,” in *International Conference on Machine Learning.* PMLR, 2018, pp. 2038–2047.

[239] A. Nitanda, “Stochastic proximal gradient descent with acceleration techniques,” in *Advances in Neural Information Processing Systems*, 2014, pp. 1574–1582.

[240] H. Lin, J. Mairal, and Z. Harchaoui, “A universal catalyst for first-order optimization,” in *Advances in Neural Information Processing Systems*, 2015, pp. 3384–3392.

[241] B. Li and G. B. Giannakis, “Adaptive step sizes in variance reduction via regularization,” *arXiv preprint arXiv:1910.06532*, 2019.

[242] J. C. Dunn, “Rates of convergence for conditional gradient algorithms near singular and nonsingular extremals,” *SIAM Journal on Control and Optimization*, vol. 17, no. 2, pp. 187–211, 1979.

[243] T. Kerdreux, A. d’Aspremont, and S. Pokutta, “Projection-free optimization on uniformly convex sets,” in *International Conference on Artificial Intelligence and Statistics.* PMLR, 2021, pp. 19–27.

[244] A. Argyriou, R. Foygel, and N. Srebro, “Sparse prediction with the k-support norm,” in *Advances in Neural Information Processing Systems*, 2012, pp. 1457–1465.

[245] L. Ding, Y. Fei, Q. Xu, and C. Yang, “Spectral Frank-Wolfe algorithm: Strict complementarity and linear convergence,” in *International Conference on Machine Learning.* PMLR, 2020, pp. 2535–2544.

[246] D. Garber, “Revisiting Frank-Wolfe for polytopes: Strict complementarity and sparsity,” in *Advances in Neural Information Processing Systems*, 2020, pp. 18 883–18 893.

[247] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[248] B. Liu, X.-T. Yuan, S. Zhang, Q. Liu, and D. N. Metaxas, "Efficient k-support-norm regularized minimization via fully corrective Frank-Wolfe method." in *International Joint Conference on Artificial Intelligence*, 2016, pp. 1760–1766.

[249] J. Bennett and S. Lanning, "The Netflix prize," in *KDD Cup and Workshop*. New York, NY, USA., 2007, p. 35.

[250] R. M. Bell and Y. Koren, "Lessons from the Netflix prize challenge," *ACM SIGKDD Explorations Newsletter*, vol. 9, no. 2, pp. 75–79, 2007.

[251] M. Fazel, "Matrix rank minimization with applications," Ph.D. dissertation, Stanford University, 2015.