Momentum for the Frank Wolfe Method

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Bingcong Li

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Professor Georgios B. Giannakis, Advisor
Professor Mingyi Hong
Professor Andrew Lamperski
Professor Ju Sun

May, 2022

Acknowledgements

There are so many people to whom I wish to express my warmest gratitude for making my past years at the University of Minnesota (UMN) the most enjoyable journey of my life.

First and foremost, my deepest gratitude goes to my Ph.D advisor Prof. Georgios B. Giannakis for his valuable guidance on every perspective of life, especially on research and career. Thanks to his incisive foresight and suggestions, I was devoted to working on the area of optimization for machine learning and signal processing problems, which constitutes the main threads of this dissertation. His guidance and constant encouragement have made me become not only a better researcher, but also a better person. Another joyful experience of being a SPiN-COMer is that you can get a cup of coffee when you walk by Prof. Giannakis' office. He also recommended me various brands of high-quality coffee beans that saved my sleepy afternoons.

Special thanks go to Professor Mingyi Hong, Professor Andrew Lamperski, and Professor Ju Sun for serving on my Ph.D committee. Professor Mingyi Hong and Professor Andrew Lamperski are two professors who I met at the very beginning of my graduate studies in UMN. Both of them are supportive and thoughtful for providing valuable suggestions on my courses and research. Professor Ju Sun has done excellent works, which have great impact on my own research even before he joined UMN.

The work in this dissertation would not have been possible without the help of my talented collaborators, who provided insightful observations, in-depth suggestions, inspiring discussions, and valuable criticism. It is my great honor to learn from and work with each of you. In particular, I'd like to extend my gratitude to Prof. Xin Wang, Prof. Tianyi Chen, Prof. Zhizhen Zhao, and Prof. Geert Leus, Dr. Meng Ma, Dr. Mario Coutino, Dr. Qin Lu, Dr. Shuai Zheng, Dr. Alireza Sadeghi, Dr. Parameswaran Raman, Dr. Jun Sun, Lingda Wang, Huozhi Zhou, Yilang Zhang, and Konstantinos Polyzos. I am truly grateful to these people for their continuous help. My special gratitude goes to my undergraduate advisor, Prof. Xin Wang, who

encouraged me to pursue a Ph.D program. Prof. Wang also provided me advices during my early Ph.D years which greatly helped my transition from an undergraduate to a researcher. I'd also thank Prof. Tianyi Chen, from whole I learned how to research during my early Ph.D years.

I also appreciate knowing and working with every SPiNCOMer, Prof. Yanning Shen, Prof. Jia Chen, Dr. Qiuling Yang, Dr. Fatemeh Sheikholeslami, Dr. Dimitris Berberidis, Dr. Donghoon Lee, Dr. Panagiotis Traganitis, Dr. Liang Zhang, Dr. Georgios V. Karanikolas, Dr. Vassilis Ioannidis, Dr. Jia Yan, Dr. Athanasios N. Nikolakopoulos, Dr. Manish Singh, Seth Barrash, Xinhu Zheng, and Shijian Gao. Life would be much more boring and tedious without SPiNCOMers. In addition, the presentation delivered by each of you in reading groups significantly broadened the scope of my research database and toolboxes. In addition, I would also like to thank Prof. George Karypis and his lab members, including Ancy Tom, Zeren Shui, Yilong Wang, Dr. Saurav Manchanda, and Costas Mavromatis for sharing the 4th floor in Walter Library. Last but not least, I would like to thank my family for all their unconditional support along the way. Your love has been and will always be invaluable to me.

Bingcong Li
Minneapolis, Feb 2022

Abstract

Modern machine learning tasks built to learn from data can be typically formulated as optimization problems. The large volume of data justifies the pressing need for efficient and scalable iterative algorithms that are designed specifically to accommodate to the computation resource at hand and the requirement of structural (e.g., sparse) solutions. Conditional gradient, aka Frank Wolfe (FW) algorithms, have well-documented merits in machine learning and signal processing applications that involves minimizing a loss function with constraints. Compared to projection based methods, one of the key benefits is that FW overcomes the need of projection, which is computationally heavy. Unlike projection-based methods however, momentum cannot improve the convergence rate of FW, in general. For this reason, momentum is relatively less studied in the FW literature. This limitation motivates the work in this dissertation.

In Chapter 2, we deal with heavy ball momentum and its impact to FW. Specifically, it is established that heavy ball offers a unifying perspective on the primal-dual (PD) convergence, and enjoys a tighter *per iteration* PD error rate, for multiple choices of step sizes, where PD error can serve as the stopping criterion in practice. In addition, it is asserted that restart, a scheme typically employed jointly with Nesterov's momentum, can further tighten this PD error bound.

Going beyond heavy ball momentum, we establish the connections between the subproblem in FW and Nesterov's momentum in Chapter 3. On the negative side, these connections show why momentum is unlikely to be effective for FW type algorithms on general problems. The encouraging message behind this link, on the other hand, is that Nesterov's momentum accelerates FW on a class of problems encountered in many signal processing and machine learning applications. In particular, we prove that a momentum variant of FW, that we term accelerated Frank Wolfe (AFW), converges with a faster rate $\mathcal{O}(\frac{1}{k^2})$ on such a family of problems despite the same $\mathcal{O}(\frac{1}{k})$ rate as FW on general cases. Our faster rates rely on parameter-free step sizes, which distinguishes with most of existing faster rates of FW variants.

Chapter 4 introduces and analyzes a variant of FW termed ExtraFW. The distinct feature of ExtraFW is the pair of gradients leveraged per iteration, thanks to which the decision variable is updated in a prediction-correction (PC) format. Relying on no problem dependent parameters in the step sizes, ExtraFW convergences at a faster rate $\mathcal{O}\big(\frac{1}{k^2}\big)$ on a class of machine learning problems. Compared with other parameter-free FW variants that have faster rates on the same problems such as AFW, ExtraFW has improved rates and fine-grained analysis thanks to its PC update.

Numerical tests on binary classification with different sparsity-promoting constraints demonstrate that the empirical performance of HFW, AFW and ExtraFW is significantly better than FW. We also observe that AFW and ExtraFW are even faster than Nesterov's accelerated gradient on certain datasets, even though they rely on no problem dependent parameters. For matrix completion, the solutions found by HFW, AFW and ExtraFW enjoy smaller optimality gap, and lower rank than FW.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The quick evolution and widespread applicability of machine learning and artificial intelligence have fundamentally reshaped and transcended people's life. From online shopping to autonomous driving cars, from smart grid to intelligent infrastructure, from social media to healthcare, learning and intelligence enabled applications and products have permeated our daily life in various ways. Two key players stand behind such a ubiquitous emergence: big data and advanced algorithms. The unprecedented amount of data generated every day lays the foundation for such a tremendous transformation. Underlying many machine learning tasks are data-driven optimization problems to be solved. The "data deluge" give rises to the need for reducing computational burden for iterative optimization solvers, as well as the favor of structured solutions (such as sparse vectors or low rank matrices) that are not only cheap to store, but also required by real-world tasks such as compressive sensing and recommender systems. Modern optimizers such as projected gradient descent (GD) built to exploit such a huge amount of data are often computationally "hungry" and their "appetite" for computing power does not endow them with the capability to directly hunt for structured solutions. All these considerations justify the pressing need for optimization algorithms that are lightweight yet flexible to promote structural solutions.

Among commonly adopted first-order methods for constrained problems, the Frank Wolfe (FW) algorithm (also known as conditional gradient method) [1] reduces the computational burden by substituting the projection step to a subproblem that can be solved much more easily. In addition, the update of FW directly promotes structural solution when sparse or low rank is of interest. FW is thus more competitive iterative solver than its projection counter part GD.

This thesis starts with theoretical perspectives of FW for convex optimization and provide a unified framework, which offers not only valuable insights, but also guidances on how to develop novel algorithms to meet emerging requirements. The vision is to broaden conventional view of FW while maintaining FW's lightweight computation, insightful geometrical explanation, and simple implementation to provide a principled means of analyzing, designing, optimizing and accelerating constrained convex problems. The ultimate goal is to guide real-world applications with supportive theories and promising empirical performance.

Notation. Bold lowercase (capital) letters denote column vectors (matrices); $\|\mathbf{x}\|$ stands for a norm of a vector $\mathbf{x}$, whose dual norm is denoted by $\|\mathbf{x}\|_*$; and $\langle \mathbf{x}, \mathbf{y} \rangle$ is the inner product of $\mathbf{x}$ and $\mathbf{y}$.

1.1 Problem statement

The Frank Wolfe (FW) method [1, 2, 3, 4, 5] is designed for solving the following constrained problem

$$\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}). \tag{1.1}$$

Here, f is the loss function, and the constraint set $\mathcal{X} \subset \mathbb{R}^d$ is assumed convex and compact, where d is the dimension of variable $\mathbf{x}$. Throughout, we let $\mathbf{x}^* \in \mathcal{X}$ denote a minimizer of (1.1). The following standard assumptions will be taken to hold true throughout.

Assumption 1. *(Lipschitz continuous gradient.) The objective function $f : \mathcal{X} \to \mathbb{R}$ has L-Lipchitz continuous gradients; i.e., $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_* \leq L\|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$.*

Assumption 2. *(Convexity.) The objective function $f : \mathcal{X} \to \mathbb{R}$ 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 3. *(Convex and compact constraint set.) The constraint set $\mathcal{X} \subset \mathbb{R}^d$ is convex and compact with diameter D, that is, $\|\mathbf{x} - \mathbf{y}\| \leq D, \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$.*

FW for solving (1.1) under Assumptions 1 – 3 is listed in Alg. 1. Although nonconvex problems can also be coped via FW [6, 7], they are beyond the scope of this thesis. The FW algorithm is simple and neat, where a FW subproblem (in line 3) and an update step (line 4) are carried out in order per iteration.

Algorithm 1 FW [1]

```
1: Initialize: $\mathbf{x}_0 \in \mathcal{X}$, $\eta_k \in [0,1],\ \forall k$
2: for $k = 0, 1, \ldots, K-1$ do
3:     $\mathbf{v}_{k+1} = \arg\min_{\mathbf{v}\in\mathcal{X}} \langle \nabla f(\mathbf{x}_k), \mathbf{v} \rangle$
4:     $\mathbf{x}_{k+1} = (1-\eta_k)\mathbf{x}_k + \eta_k \mathbf{v}_{k+1}$
5: end for
6: Return: $\mathbf{x}_K$
```

In various machine learning and signal processing applications, the FW subproblem in line 3 is much easier to be solved compared with projection steps in GD, as one can see that the loss function of FW subproblem is linear. The popularity of FW is partially due to the elimination of projection compared with projected gradient descent (GD) [8], leading to computational efficiency especially when d is large. Next, we will provide a few examples of applications, and then discuss the supporting theories for FW.

1.2 Use cases of FW

This section provides several examples of scenarios that favors the adoption of FW (variants).

1.2.1 Sparse signal recovery

Suppose that we want to recover a signal $\mathbf{x}^* \in \mathbb{R}^n$ as a sparse representation of observations $\mathbf{y} = \mathbf{A}\mathbf{x}^* + \mathbf{e}$, where $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{e} \sim \mathcal{N}(0, \sigma^2\mathbf{I}_m)$ is the stochastic noise. One would naturally formulate the problem as

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \\ \text{s.t.} \quad & \|\mathbf{x}\|_0 \leq \|\mathbf{x}^*\|_0. \end{aligned}$$

However, the nonconvex constraint, i.e, ℓ_0-norm ball, renders the problem intractable in many situations. The most widely adopted remedy is to use ℓ_1-norm as a surrogate to the ℓ_0-norm ball [9], leading the problem to

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \\ \text{s.t.} \quad & \|\mathbf{x}\|_1 \leq \|\mathbf{x}^*\|_1. \end{aligned}$$

In practice, $\|\mathbf{x}^*\|_1 := R$ is unknown and is typically chosen by cross validation. It is not hard to verify that Assumptions 1 – 3 hold for this problem, and hence FW can be applied directly.

Sparsity promoting property of FW for ℓ_1-norm ball constraint. Unlike projection based algorithms such as GD, FW in Alg. 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 that $k-1$ entries are non-zero in $\mathbf{x}_{k-1}$.

1.2.2 Sparse classification

Sparse classification is a central problem in machine learning as it leads to more interpretable and robust models [10]. There are different ways to formulate this problem, here we adopt logistic regression for binary classification as an example. The problem is formulated as

$$\begin{aligned} \min_{\mathbf{x}} \;\; & \frac{1}{N}\sum_{i=1}^{N} \ln\big(1+\exp(-b_i\langle \mathbf{a}_i, \mathbf{x}\rangle)\big) \\ \text{s.t.} \;\; & \|\mathbf{x}\|_0 \leq k. \end{aligned}$$

Here $(\mathbf{a}_i, b_i)$ is the (feature, label) pair of datum i, N is the number of data, and k is the expected number of 0s in the classifier. Similar to sparse signal recovery, ℓ_1-norm is more favorable as an approximation to the ℓ_0-norm ball, which further leads to the problem as

$$\begin{aligned} \min_{\mathbf{x}} \;\; & \frac{1}{N}\sum_{i=1}^{N} \ln\big(1+\exp(-b_i\langle \mathbf{a}_i, \mathbf{x}\rangle)\big) \\ \text{s.t.} \;\; & \|\mathbf{x}\|_1 \leq R. \end{aligned}$$

In practice, R is unknown and is typically chosen by cross validation. FW also promotes a sparse solution for this problem.

Other constraints. The ℓ_1 norm ball in the aforementioned problem can be substituted to more sophisticated constraints, e.g., ordered weighted ℓ_1 norm (OWL) [11], and n-support norm ball [12], to promote a sparse solution as well. In such cases, the FW subproblem is also cheaper than projection.

1.2.3 Matrix completion

Matrix completion problems (or collaborative filtering) appear widely in recommender systems. Consider a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ with partially observed entries, i.e., entries A_{ij} for $(i,j) \in \mathcal{K}$ are known, where $\mathcal{K} \subset \{1,\dots,m\} \times \{1,\dots,n\}$, where A_{ij} is the rating of user i to item j. Based on the observed entries that can be contaminated by noise, the goal is to predict the missing ones. Within the scope of recommender systems, a commonly adopted empirical observation is that $\mathbf{A}$ is low rank [13, 14, 15], leading to the following problem formulation.

$$\begin{aligned} \min_{\mathbf{X}} \ & \frac{1}{2} \sum_{(i,j)\in\mathcal{K}} (X_{ij} - A_{ij})^2 \\ \text{s.t.} \ & \|\mathbf{X}\|_{\text{nuc}} \leq R. \end{aligned} \tag{1.2}$$

Problem (1.2) is difficult to solve using GD because projection onto a nuclear norm ball requires a full SVD, which has complexity $\mathcal{O}\big(mn(m \wedge n)\big)$ with $(m \wedge n) := \min\{m,n\}$. In contrast, FW and its variants are more suitable for (1.2) since the FW subproblem has complexity less than $\mathcal{O}(mn)$ [16].

Low rank promoting property of FW under nuclear norm ball constraint. In addition to the projection-free property, FW is more suitable for this problem compared to GD because it also guarantees $\text{rank}(\mathbf{X}_k) \leq k+1$ [17, 18]. Suppose that the singular value decomposition (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 subproblem can be solved easily by

$$\boldsymbol{V}_{k+1} = -R\mathbf{p}_k\mathbf{q}_k^\top \tag{1.3}$$

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 (1.3) has rank at most 1. Hence it is easy to see $\mathbf{X}_{k+1} = (1-\delta_k)\mathbf{X}_k + \delta_k \boldsymbol{V}_{k+1}$ has rank at most $k+2$ if $\mathbf{X}_k$ is a rank-$(k+1)$ matrix (i.e., $\mathbf{X}_0$ has rank 1).

1.2.4 Coordination of electric vehicle charging

The convex setup of optimal schedules for electric vehicle (EV) charging in [19] is briefly reviewed next. Suppose that a load aggregator coordinates the charging of N EVs over the T

consecutive time slots $\mathcal{T} := 1, ..., T$ of length Δ_τ. Let $\mathcal{T}_n \subseteq \mathcal{T}$ denote the time slots in which vehicle n is connected to the power grid, and let $x_n(\tau)$ be the charging rate of EV n at time τ to be scheduled by the load aggregator. If $\bar{x}_n$ is the charging rate limitation imposed by the battery of vehicle n, then $x_n(\tau)$ should lie in the interval $[0, \bar{x}_n]$ with

$$\bar{x}_n(\tau) := \begin{cases} \bar{x}_n, & \tau \in \mathcal{T}_n \\ 0, & \text{otherwise} \end{cases}$$

The charging profile for vehicle n, denoted by $\mathbf{x}_n^\top := [x_n(1), \ldots, x_n(T)]$ should therefore belong to the convex and compact set

$$\mathcal{X}_n := \{\mathbf{x}_n | \Delta_\tau \mathbf{x}_n^\top \mathbf{1} = R_n,\ \ 0 \leq x_n(\tau) \leq \bar{x}_n(\tau),\ \forall \tau \in \mathcal{T}\} \tag{1.4}$$

where R_n represents the total energy needed by EV n.

Given $\{R_n\}_{n=1}^N$, $\{x_n\}_{n=1}^N$, and $\{\mathcal{T}_n\}_{n=1}^N$, the problem solved by the aggregator is to find the charging profiles minimizing its electricity cost

$$\begin{aligned} \min_{\mathbf{x}} \ \ & f(\mathbf{x}) \\ \text{s.t.} \ \ & \mathbf{x}_n \in \mathcal{X}_n,\ \forall n \in \mathcal{N} \end{aligned}$$

where $f(\mathbf{x})$ should is customizable and chosen as $f(\mathbf{x}) = \sum_{\tau=1}^T \left(\sum_{n=1}^N x_n(\tau) + D(\tau) \right)^2$ in [20], where $D(\tau)$ denotes additional known loads.

As the constraint set is affine, the problem can be solved via FW, where a closed form solution exists for the FW subproblem. Noticing the blockwise structure of the constraint set, i.e., the feasible set is the Cartesian product $\mathcal{X} := \mathcal{X}_1 \times \ldots \times \ldots \mathcal{X}_N$, this problem also favors the adoption of a specifically designed FW variant, randomized block FW (RB-FW) [20, 21], to accommodate the block-wise structure in $\mathcal{X}$, especially when N is large.

1.2.5 Neural network pruning

Deep neural networks (DNNs) enjoy documented success in real-world tasks that emerge from various applications, including computer vision [22], and natural language processing [23]. The widespread consensus however, is that the 'resource-hungry' DNNs are not yet ready to undertake several tasks of the emerging Internet of Things (IoT) [24, 25], where devices can have stringent computation and memory constraints [26]. With the success of transformers having

0.6B parameters in vision tasks [27], it is foreseeable that the model size will quickly scale up to achieve improved performance on even more sophisticated tasks. However, these models of overwhelming size are difficult to learn using embedded systems, such as those in autonomous driving cars and smartphones. These considerations coupled with empirical observations that DNNs are often highly redundant, prompts the possibility to remove unnecessary neurons while minimally sacrificing accuracy – what DNN pruning promises to accomplish.

Consider for brevity a two-layer NN, although pruning pertains also to multi-layer DNNs. With slightly abused notation, let $\mathbf{x}$ and y denote respectively the feature and label of a training datum; $\sigma_i(\cdot)$ the n-th neuron's memoryless nonlinearity, e.g., $\sigma_i(\mathbf{x}) := \mathbf{w}_{i2}\mathrm{ReLU}(\mathbf{w}_{i1}^{\top}\mathbf{x})$; and $\{\mathbf{w}_{i1}, \mathbf{w}_{i2}\}$ the weights to be learned. The two-layer NN with N neurons models the map $\mathbf{x} \to y$, using the function

$$f_N(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\sigma_i(\mathbf{x}) \; . \tag{1.5}$$

Although several options are possible, suppose that training relies on minimizing the square loss

$$\mathcal{L}^* := \frac{1}{2}\mathbb{E}_{(\mathbf{x},\mathbf{y})}\left[\|f_N^*(\mathbf{x}) - y\|^2\right] := \min_{\{\mathbf{w}_{i1},\mathbf{w}_{i2}\}_i}\frac{1}{2}\mathbb{E}_{(\mathbf{x},y)}\left[\|f_N(\mathbf{x}) - y\|^2\right] \tag{1.6}$$

where $f_N^*(\mathbf{x})$ is the NN map with optimally trained weights, and $\sigma_i^*(\mathbf{x})$ the i-th neuron output.

Having obtained $f_N^*(\mathbf{x})$, the pruning task targets at a subnetwork $\mathcal{S}$ of cardinality $|\mathcal{S}| < n$ whose neurons are judiciously selected from $f_N^*(\mathbf{x})$, and with the pruned NN effecting the map $f_{\mathcal{S}}^*(\mathbf{x}) := \frac{1}{N}\sum_{n=1}^{N}\sigma_n^*(\mathbf{x})\mathbb{1}(n \in \mathcal{S})$, where $\mathbb{1}$ denotes the binary indicator function. As neurons with low impact on the training loss must be pruned, $f_{\mathcal{S}}^*(\mathbf{x})$ solves the optimization problem

$$\min_{|\mathcal{S}|<n}\mathcal{L}(\mathcal{S}) := \frac{1}{2}\mathbb{E}_{(\mathbf{x},\mathbf{y})}\left[\|f_{\mathcal{S}}^*(\mathbf{x}) - y\|^2\right]. \tag{1.7}$$

The combinatorial complexity of solving (1.7) explains why several existing works only provide heuristic methods [28, 29, 30, 31]. Another critical yet time-consuming task is that of tuning for the best n to account for the tradeoff of lowering the pruning loss and sparsifying the subnetwork. Applying FW based approaches will systematically cope with these two concerns, and provide guidelines for pruned DNNs with guaranteed robustness.

A desirable pruning method should remove as many as possible unnecessary neurons in a pre-trained DNN, while incurring minimal pruning-induced training loss. A neat link bridging

DNN pruning with the FW was revealed in [32]. Per FW iteration, a new neuron obtained by solving the FW subproblem, is added to the target set $\mathcal{S}$. It can be established that the FW method guarantees the training loss $\mathcal{L}(\mathcal{S})$ of the pruned DNN with n neurons left to be no larger than $\mathcal{L}^* + \epsilon_n$, where $\epsilon_n = \mathcal{O}(\frac{\ln n}{n})$ [32]. This result readily quantifies the worst-case performance of the pruned NN relative to that of the original unpruned NN, hence ensuring robustness.

1.2.6 Other applications

Besides aforementioned tasks, other tasks that favors FW or its variants include e.g., traffic assignment [33], non-negative matrix factorization [34], video colocation [35], image reconstruction [17], particle filtering [36], cluster detection in networks [37], adversarial training of neural networks [38], tuning step sizes for training large neural networks [39], and optimal transport [40].

1.3 FW theories

In this section, theories behind FW are briefly recapped. As in most optimization algorithms, two essential ingredients are the update direction and the step sizes. Rewriting line 4 of Alg. 1 as $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k(\mathbf{x}_k - \mathbf{v}_{k+1})$, it is clear the update direction is $-(\mathbf{x}_k - \mathbf{v}_{k+1})$ with step size η_k. More details on the update direction are reviewed next.

1.3.1 A close view of the FW subproblem

FW subproblem ensures a descent direction. Since $\mathbf{v}_{k+1}$ minimizes $\langle \nabla f(\mathbf{x}_k), \mathbf{v}\rangle$ over $\mathcal{X}$, we have

$$\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k\rangle \leq 0.$$

This inequality implies that the update direction in FW is a descent direction.

Geometry of FW subproblem. The subproblem in line 3 of Alg. 1 can be visualized geometrically as minimizing a supporting hyperplane of $f(\mathbf{x})$ at $\mathbf{x}_k$, i.e.,

$$\mathbf{v}_{k+1} \in \arg\min_{\mathbf{v}\in\mathcal{X}} f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{v} - \mathbf{x}_k\rangle. \tag{1.8}$$

Upon minimizing the supporting hyperplane in (1.8), $\mathbf{x}_{k+1}$ is updated as a convex combination of $\mathbf{v}_{k+1}$ and $\mathbf{x}_k$ in line 4 so that no projection is required. The choices on the step size $\eta_k \in [0, 1]$ will be discussed shortly.

The hyperplane geometry behind FW subproblem enables the calculation of primal-dual (PD) error or the so-termed *FW gap*, formally defined as

$$\begin{aligned}\bar{\mathcal{G}}_k &:= \langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{v}_{k+1} \rangle \\ &= \underbrace{f(\mathbf{x}_k) - f(\mathbf{x}^*)}_{\text{primal error}} + \underbrace{f(\mathbf{x}^*) - \min_{\mathbf{v}\in\mathcal{X}} \Big[f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{v} - \mathbf{x}_k \rangle \Big]}_{\text{dual error}} \end{aligned} \tag{1.9}$$

where the second equation is because of (1.8). It can be verified that both primal and dual errors marked in (1.9) are no less than 0 by appealing to the convexity of f. If $\bar{\mathcal{G}}_k$ converges, one can deduce that the primal error converges. For this reason, $\bar{\mathcal{G}}_k$ is typically used as a stopping criterion for Alg. 1.

Complexity of FW subproblem. For many constraint sets, efficient implementation or a closed-form solution is available for $\mathbf{v}_{k+1}$, a few examples are provided in Table. 1.1, where N_f denotes the number of non-zero entries of the gradient; ϵ denotes the desirable accuracy if one has to use an iterative solver for FW subproblem; and $\sigma_1(.)$ is the largest singular value. Please refer to [2, 41] for more comprehensive summaries of constraints on which FW subproblem can be solved efficiently.

Table 1.1: FW subproblem complexity

Constraint	solution	complexity
$\big\{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|_1 \leq 1\big\}$	Holder inequality	d
$\big\{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|_\infty \leq 1\big\}$	Holder inequality	d
$\big\{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|_p \leq 1\big\}, p \in (1, \infty)$	Holder inequality	d
$\big\{\mathbf{X} \in \mathbb{R}^{m\times n} \mid \|\mathbf{X}\|_{\text{nuc}} \leq 1\big\}$	Relate to $\sigma_1(\nabla f(\mathbf{X}))$	$\mathcal{O}(N_f)$
$\big\{\mathbf{X} \in \mathbb{S}^{n\times n} \mid \mathbf{X} \succeq \mathbf{0}, \text{Tr}(\mathbf{X}) = 1\big\}$	Lanzcos alg.	$\tilde{\mathcal{O}}(N_f/\epsilon)$

Other update directions. While the standard update direction is universal, there are more efficient ones for specific problems, typically when $\mathcal{X}$ is a polytope and $f(\mathbf{x})$ is strongly convex [42]. Such directions can lead to faster convergence. For example, linear rate can be achieved using the aid of away-steps [43]. The idea is that in each iteration, we not only add a new vertex $\mathbf{s}$, but potentially also remove an old vertex (provided it is bad with respect to our objective). Besides away-steps, other update directions include e.g., in-face directions [18], pairwise directions [44, 42].

1.3.2 FW step sizes

Choosing the step size $\eta_k \in [0, 1]$, it is clear that $\mathbf{x}_{k+1}$ is a convex combination of $\mathbf{x}_k$ and $\mathbf{v}_{k+1}$. Hence, long as $\mathbf{x}_k \in \mathcal{X}$, $\mathbf{x}_{k+1}$ will also lives in $\mathcal{X}$. Next, we outline three commonly used step sizes in the literature.

Parameter-free step size. This type of step sizes does not rely on any problem dependent parameters such as L and D, and hence it is extremely simple to implement. The most commonly adopted step size is $\eta_k = \frac{2}{k+2}$, which ensures a converging primal error $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{2LD^2}{k+1}, \forall k \geq 1$, and a weaker claim on the PD error, $\min_{k \in \{1,\ldots,K\}} \bar{\mathcal{G}}_k = \frac{27LD^2}{4K}$ [2]. One can also choose $\eta_k = \frac{1}{k+1}$, which leads to a primal error of $\mathcal{O}(\frac{LD^2 \ln k}{k})$.

Smooth step size. When the (estimate of) Lipschitz constant L is available, one can adopt the following step sizes in Alg. 1 [45]

$$\eta_k = \min\left\{\frac{\langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{v}_{k+1}\rangle}{L\|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}, 1\right\}. \tag{1.10}$$

Despite the estimated L is typically too pessimistic to capture the local Lipschitz continuity, such a step size ensures $f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k)$. To see this, we have from Assumption 1 that

$$\begin{aligned} f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) &\leq \langle \nabla f(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k\rangle + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \qquad (1.11) \\ &\overset{(a)}{=} \eta_k \langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k\rangle + \frac{\eta_k^2 L}{2}\|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2 \overset{(b)}{\leq} 0 \end{aligned}$$

where (a) uses $\mathbf{x}_{k+1} = (1 - \eta_k)\mathbf{x}_k + \eta_k \mathbf{v}_{k+1}$; and (b) is because η_k minimizes the RHS of (1.11) over $[0, 1]$.

Line search. We can also choose the step size η_k via line search, although this might be more computationally costly in practice because it requires computing the function value. The parameters are selected as

$$\eta_k = \arg\min_{\eta \in [0,1]} f\big((1 - \eta)\mathbf{x}_k + \eta \mathbf{v}_{k+1}\big). \tag{1.12}$$

Such a parameter choice also ensures per step objective descent since

$$f(\mathbf{x}_{k+1}) = \min_{\eta\in[0,1]} f\big((1-\eta)\mathbf{x}_k + \eta\mathbf{v}_{k+1}\big)$$
$$\overset{(a)}{\leq} f\big((1-\theta)\mathbf{x}_k + \theta\mathbf{v}_{k+1}\big) \overset{(b)}{=} f(\mathbf{x}_k)$$

where in (a) we have $\theta \in [0, 1]$; and in (b) we set $\theta = 0$.

Convergence rate. It has been established that for all aforementioned step sizes, the primal error, $f(\mathbf{x}_k) - f(\mathbf{x}^*)$, converges at a rate of $\mathcal{O}(\frac{LD^2}{k})$. Regarding the primal-dual error, existing works are not satisfactory, and this will be discussed in detail in Chapter 2.

Existing literature also relies on blackbox optimization paradigm, where the objective function and constraint set can be accessed through oracles only. For FW in particular, the first-order oracle (FO) and the linear minimization oracle (LMO) are needed.

Definition 1. *(FO.) The first-order oracle takes* $\mathbf{x} \in \mathcal{X}$ *as an input and returns its gradient* $\nabla f(\mathbf{x})$.

Definition 2. *(LMO.) The linear minimization oracle takes a vector* $\mathbf{g} \in \mathbb{R}^d$ *as an input and returns a minimizer of* $\min_{\mathbf{x}\in\mathcal{X}}\langle \mathbf{g}, \mathbf{x}\rangle$.

The efficiency of an algorithm in blackbox optimization paradigm is characterized by the number of oracles used to achieve the desirable primal error $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \epsilon$. Since each FW iteration requires one FO and one LMO, simply setting convergence rate smaller than ϵ, the oracle complexity for FW, i.e., the number of oracles needed, can be calculated as $\mathcal{O}(\frac{LD^2}{\epsilon})$. Since the relation between convergence rate and oracle complexity is straightforward, in this thesis we use these two terms interchangeably.

1.4 Open issues of FW and contributions

While FW has well-documented merits, there are several open problems that can further benefit FW type algorithms. Among the unaddressed issues, the work in this thesis would mainly deal FW with momentum, including heavy ball (aka Polyak's) momentum and Nesteorv's momentum.

While momentum is ubiquitous in projection based algorithms for obtaining either theoretical or numerical benefits [46, 47, 48], it is known that momentum does not perform well with FW. Indeed, the lower bound in [2, 3] demonstrates that at least $\mathcal{O}(\frac{LD^2}{\epsilon})$ LMO calls are required to ensure $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \epsilon$, which does not guarantee that momentum is beneficial for FW, because even vanilla FW achieves this lower bound. Due to this negative result, momentum is not carefully studied in FW literature. This thesis will systematically study the use of momentum in FW, and provide provable evidences on the usefulness of momentum.

Chapter 2

Heavy ball momentum for FW

2.1 Introduction

In this chapter, we contend that momentum is evidently useful for FW. Specifically, we prove that the *heavy ball momentum* leads to tightened and efficiently computed primal-dual error bound, as well as numerical improvement. To this end, we outline first the primal convergence.

Primal convergence. The primal error refers to $f(\mathbf{x}_k) - f(\mathbf{x}^*)$. It is guaranteed for FW that $f(\mathbf{x}_k) - f(\mathbf{x}^*) = \mathcal{O}\big(LD^2/k\big), \forall k \geq 1$ [2, 45]. This rate is tight in general since it matches to the lower bound [3, 2]. Other FW variants also ensure the same order of primal error; see e.g., [3, 49].

Primal-dual convergence. The primal-dual (PD) error quantifies the difference between both the primal and the 'dual' functions from the optimal objective, hence it is an upper bound on the primal error. When the PD error is shown to *converge*, it can be safely used as the stopping criterion: whenever the PD error is less than some prescribed $\epsilon > 0$, $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \epsilon$ is ensured automatically. The PD error of FW is convenient to compute, hence FW is suitable for the requirement of "solving problems to some desirable accuracy;" see e.g., [50]. For pruning (two-layer) neural networks [32], the extra training loss incurred by removing neurons can be estimated via the PD error. However, due to technical difficulties, existing analyses on PD error are not satisfactory enough and lack of unification. It is established in [51, 2, 4] that the minimum PD error is sufficiently small, namely $\min_{k \in \{1,\dots,K\}} \text{PDError}_k = \mathcal{O}\big(\frac{LD^2}{K}\big)$, where K is the total number of iterations. We term such a bound for the minimum PD error as Type I guarantee. Another stronger guarantee, which directly implies Type I bound, emphasizes the per iteration convergence, e.g., $\text{PDError}_k \leq \mathcal{O}(\frac{LD^2}{k}), \forall k$. We term such guarantees as Type II bound. A Type II bound is reported in [42, Theorem 2], but with an unsatisfactory k dependence. This is improved by [52, 53] with the price of extra computational burden since it involves solving *two* FW subproblems per iteration for computing this PD error. Several related works such as [4] provide a weaker PD error compared with [52]; see a summary in Table 3.1.

Table 2.1: A comparison of HFW with relevant works. The "computation" in the third column is short for "the number of required FW subproblems to calculate the PD error per iteration."

reference	**computation**	**PD conv. type**	**PD conv. rate**
[2]	1 subproblem	Type I	$\frac{27LD^2}{4(K+1)}$
[42]	2 subproblems	Type II	$\frac{2LD^2}{\sqrt{k+1}}, \forall k$
[52]	2 subproblems	Type II	$\frac{4LD^2}{k+1}, \forall k$
This work (Alg. 2)	1 subproblem	Type II	$\frac{2LD^2}{k+1}, \forall k$
This work (Alg. 3)	2 subproblems	Type II	$\frac{2LD^2}{k+1+c}, \forall k$ with $c \geq 0$

In this chapter, we show that a computationally affordable Type II bound can be obtained by simply relying on heavy ball momentum. Interestingly, FW based on heavy ball momentum (HFW) also maintains FW's neat geometric interpretation. Through unified analysis, the resultant type II PD error improves over existing bounds; see Table 1. This PD error of HFW is further tightened using *restart*. Although restart is more popular in projection based methods together with Nesterov's momentum [54], we show that restart for FW is natural to adopt jointly with heavy ball. In succinct form, our contributions can be summarized as follows.

- We show through unified analysis that HFW enables a tighter type II guarantee for PD error for multiple choices of the step size. When used as stopping criterion, no extra subproblem is needed.

- The Type II bound can be further tightened by restart triggered through a comparison between two PD-error-related quantities.

- Numerical tests on benchmark datasets support the effectiveness of heavy ball momentum. As a byproduct, a simple yet efficient means of computing local Lipschitz constants becomes available to improve the numerical efficiency of smooth step sizes [45, 55].

2.2 FW with heavy ball momentum

This section focuses on the benefits of heavy ball momentum for FW under multiple step size choices, with special emphasis on PD errors.

2.2.1 Algorithm

Algorithm 2 FW with heavy ball momentum (HFW)

```
**Initialize:** $\mathbf{x}_0 \in \mathcal{X}, \mathbf{g}_0 = \nabla f(\mathbf{x}_0)$
**for** $k = 0, 1, \ldots, K-1$ **do**
    $\mathbf{g}_{k+1} = (1-\delta_k)\mathbf{g}_k + \delta_k \nabla f(\mathbf{x}_k)$
    $\mathbf{v}_{k+1} = \arg\min_{\mathbf{v}\in\mathcal{X}} \langle \mathbf{g}_{k+1}, \mathbf{v}\rangle$
    $\mathbf{x}_{k+1} = (1-\eta_k)\mathbf{x}_k + \eta_k \mathbf{v}_{k+1}$
**end for**
**Return:** $\mathbf{x}_K$
```

HFW is summarized in Alg. 2. Similar to GD with heavy ball momentum [46, 47], Alg. 2 updates decision variables using a weighted average of gradients $\mathbf{g}_{k+1}$. In addition, the update direction of Alg. 2 is no longer guaranteed to be a descent one. This is because in HFW, $\langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{v}_{k+1}\rangle$ can be negative. Although a stochastic version of heavy ball momentum was adopted in [56] and its variants, e.g., [57], to reduce the mean square error of the gradient estimate, heavy ball is introduced here for a totally different purpose, that is, to improve the PD error. The most significant difference comes at technical perspectives, which is discussed in Sec. 2.2.5. Next, we gain some intuition on why heavy ball can be beneficial.

Consider $\mathcal{X}$ as an ℓ_2-norm ball, that is, $\mathcal{X} = \{\mathbf{x} | \|\mathbf{x}\|_2 \leq R\}$. In this case, we have $\mathbf{v}_{k+1} = -\frac{R}{\|\mathbf{g}_{k+1}\|_2}\mathbf{g}_{k+1}$ in Alg. 2. The momentum $\mathbf{g}_{k+1}$ can smooth out the changes of $\{\nabla f(\mathbf{x}_k)\}$, resulting in a more concentrated sequence $\{\mathbf{v}_{k+1}\}$. Recall that the PD error is closely related to $\mathbf{v}_{k+1}$ [cf. equation (1.9)]. We hope the "concentration" of $\{\mathbf{v}_{k+1}\}$ to be helpful in reducing the changes of PD error among consecutive iterations so that a Type II PD error bound is attainable.

A few concepts are necessary to obtain a tightened PD error of HFW. First, we introduce the generalized FW gap associated with Alg. 2 that captures the PD error. Write $\mathbf{g}_{k+1}$ explicitly as $\mathbf{g}_{k+1} = \sum_{\tau=0}^{k} w_k^\tau \nabla f(\mathbf{x}_\tau)$, where $w_k^\tau = \delta_\tau \prod_{j=\tau+1}^{k}(1-\delta_j) > 0$, $\forall \tau \geq 1$, and $w_k^0 =$

$\prod_{j=1}^{k}(1-\delta_j) > 0$. Then, define a sequence of linear functions $\{\Phi_k(\mathbf{x})\}$ as

$$\Phi_{k+1}(\mathbf{x}) := \sum_{\tau=0}^{k} w_k^\tau \big[f(\mathbf{x}_\tau) + \langle \nabla f(\mathbf{x}_\tau), \mathbf{x} - \mathbf{x}_\tau \rangle\big], \; \forall k \geq 0. \tag{2.1}$$

It is clear that $\Phi_{k+1}(\mathbf{x})$ is a weighted average of the supporting hyperplanes of $f(\mathbf{x})$ at $\{\mathbf{x}_\tau\}_{\tau=0}^{k}$. The properties of $\Phi_{k+1}(\mathbf{x})$, and how they relate to Alg. 2 are summarized in the next lemma.

Lemma 1. *For the linear function $\Phi_{k+1}(\mathbf{x})$ in (2.1), it holds that: i) $\mathbf{v}_{k+1}$ minimizes $\Phi_{k+1}(\mathbf{x})$ over $\mathcal{X}$; and, ii) $f(\mathbf{x}) \geq \Phi_{k+1}(\mathbf{x}), \forall k \geq 0, \forall \mathbf{x} \in \mathcal{X}$.*

From the last lemma, one can see that $\mathbf{v}_k$ is obtained by minimizing $\Phi_k(\mathbf{x})$, which is an affine lower bound on $f(\mathbf{x})$. Hence, HFW admits a geometric interpretation similar to that of FW. In addition, based on $\Phi_k(\mathbf{x})$ we can define the generalized FW gap.

Definition 3. *(Generalized FW gap.) The generalized FW gap w.r.t. $\Phi_k(\mathbf{x})$ is*

$$\mathcal{G}_k := f(\mathbf{x}_k) - \min_{\mathbf{x} \in \mathcal{X}} \Phi_k(\mathbf{x}) = f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k). \tag{2.2}$$

In words, the generalized FW gap is defined as the difference between $f(\mathbf{x}_k)$ and the minimal value of $\Phi_k(\mathbf{x})$ over $\mathcal{X}$. The newly defined $\mathcal{G}_k$ also illustrates the PD error

$$\mathcal{G}_k = f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k) = \underbrace{f(\mathbf{x}_k) - f(\mathbf{x}^*)}_{\text{primal error}} + \underbrace{f(\mathbf{x}^*) - \Phi_k(\mathbf{v}_k)}_{\text{dual error}}. \tag{2.3}$$

For the dual error, we have $f(\mathbf{x}^*) - \Phi_k(\mathbf{v}_k) \geq \Phi_k(\mathbf{x}^*) - \Phi_k(\mathbf{v}_k) \geq 0$, where both inequalities follow from Lemma 1. Hence, $\mathcal{G}_k \geq 0$ automatically serves as an overestimate of both primal and dual errors. When establishing the convergence of $\mathcal{G}_k$, it can be adopted as the stopping criterion for Alg. 2. Related claims have been made for the generalized FW gap [52, 3, 58]. Lack of heavy ball momentum leads to inefficiency, because an additional FW subproblem is needed to compute this gap [52]. Having defined the generalized FW gap, we next pursue parameter choices that establish Type II convergence guarantees.

2.2.2 Parameter-free step size

We first consider a parameter-free choice for HFW to demonstrate the usefulness of heavy ball

$$\delta_k = \eta_k = \frac{2}{k+2}, \; \forall k \geq 0. \tag{2.4}$$

Such a choice on δ_k puts more weight on recent gradients when calculating $\mathbf{g}_{k+1}$, since $w_k^\tau = \mathcal{O}(\frac{\tau}{k^2})$. The following theorem specifies the convergence of $\mathcal{G}_k$.

Theorem 1. *If Assumptions 1-3 hold, then choosing δ_k and η_k as in* (2.4)*, Alg. 2 guarantees that*

$$\mathcal{G}_k = f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k) \leq \frac{2LD^2}{k+1}, \ \forall k \geq 1.$$

Theorem 1 provides a much stronger PD guarantee for all k than vanilla FW [2, Theorem 2]. In addition to a readily computable generalized FW gap, our rate is tighter than [52], where the provided bound is $\frac{4LD^2}{k+1}$. In fact, the constants in our PD bound even match to the best known primal error of vanilla FW. A direct consequence of Theorem 1 is the convergence of both primal and dual errors.

Corollary 1. *Choosing the parameters as in Theorem 1, then $\forall k \geq 1$, Alg.2 guarantees that*

$$\begin{aligned} \textit{primal conv.:} \ \ f(\mathbf{x}_k) - f(\mathbf{x}^*) &\leq \frac{2LD^2}{k+1}; \\ \textit{dual conv.:} \ \ f(\mathbf{x}^*) - \Phi_k(\mathbf{v}_k) &\leq \frac{2LD^2}{k+1}. \end{aligned}$$

Proof. Combine Theorem 1 with $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \mathcal{G}_k$ and $f(\mathbf{x}^*) - \Phi_k(\mathbf{v}_k) \leq \mathcal{G}_k$ [cf. (2.3)]. □

2.2.3 Smooth step size

Next, we focus on HFW with a variant of the smooth step size

$$\begin{aligned} \delta_k &= \frac{2}{k+2} \\ \eta_k &= \max\left\{0, \min\left\{\frac{\langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{v}_{k+1}\rangle}{L\|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}, 1\right\}\right\}. \end{aligned} \tag{2.5}$$

Comparing with the smooth step size for vanilla FW in (1.10), it can be deduced that the choice on η_k in (2.5) has to be trimmed to $[0, 1]$ manually. This is because $\langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{v}_{k+1}\rangle$ is no longer guaranteed to be positive. The smooth step size enables an adaptive means of adjusting the weight for $\nabla f(\mathbf{x}_k)$. To see this, note that when $\eta_k = 0$, we have $\mathbf{x}_{k+1} = \mathbf{x}_k$. As a result, $\mathbf{g}_{k+2} = (1 - \delta_{k+1})\mathbf{g}_{k+1} + \delta_{k+1}\nabla f(\mathbf{x}_{k+1}) = (1 - \delta_{k+1})\mathbf{g}_{k+1} + \delta_{k+1}\nabla f(\mathbf{x}_k)$, that is, the weight on $\nabla f(\mathbf{x}_k)$ is adaptively increased to $\delta_k(1 - \delta_{k+1}) + \delta_{k+1}$ if one further unpacks $\mathbf{g}_{k+1}$. Another analytical benefit of the step size in (2.5) is that it guarantees a non-increasing objective value; see Appendix 2.5.1 for derivations. Convergence of the generalized FW gap is established next.

Theorem 2. *If Assumptions 1-3 hold, while η_k and δ_k are chosen as in* (2.5)*, Alg. 2 guarantees that*

$$\mathcal{G}_k = f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k) \leq \frac{2LD^2}{k+1}, \ \forall k \geq 1.$$

The proof of Theorem 2 follows from that of Theorem 1 after modifying just one inequality. This considerably simplifies the analysis on the (modified) FW gap compared to vanilla FW with smooth step size [18]. The PD convergence clearly implies the convergence of both primal and dual errors. A similar result to Corollary 1 can be obtained, but we omit it for brevity. We further extend Theorem 2 in Appendix 2.5.5 by showing that if a slightly more difficult subproblem can be solved, it is possible to ensure *per step descent on the PD error; i.e., $\mathcal{G}_{k+1} \leq \mathcal{G}_k$.*

2.2.4 Line search

We can also choose the step size η_k via line search, although this might be computationally costly in practice because it requires computing the function value multiple times. The parameters are selected as

$$\delta_k = \frac{2}{k+2}, \ \forall k \geq 0 \tag{2.6a}$$

$$\eta_k = \arg\min_{\eta \in [0,1]} f\big((1-\eta)\mathbf{x}_k + \eta \mathbf{v}_{k+1}\big). \tag{2.6b}$$

Such a parameter choice also ensures per step objective descent since

$$\begin{aligned} f(\mathbf{x}_{k+1}) &= \min_{\eta \in [0,1]} f\big((1-\eta)\mathbf{x}_k + \eta \mathbf{v}_{k+1}\big) \\ &\overset{(a)}{\leq} f\big((1-\theta)\mathbf{x}_k + \theta \mathbf{v}_{k+1}\big) \overset{(b)}{=} f(\mathbf{x}_k) \end{aligned}$$

where in (a) we have $\theta \in [0,1]$; and in (b) we set $\theta = 0$. Primal-dual convergence is established next, and the proof can be found in Appendix 2.5.6.

Theorem 3. *If Assumptions 1-3 hold, while δ_k and η_k are chosen via* (2.6)*, Alg. 2 guarantees that*

$$\mathcal{G}_k = f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k) \leq \frac{2LD^2}{k+1}, \ \forall k \geq 1.$$

For completeness, an iterative manner to update $\mathcal{G}_k$ for using as stopping criterion is also described in Appendix 2.5.10.

2.2.5 Further considerations

There are more choices of δ_k and η_k leading to (primal) convergence. For example, one can choose $\delta_k \equiv \delta \in (0,1)$ and $\eta_k = \mathcal{O}\big(\frac{1}{k}\big)$ as an extension of [56].[1] A proof is provided in Appendix 2.5.8 for completeness. This analysis framework in [56], however, has two shortcomings: i) the convergence can be only established using ℓ_2-norm (recall that in Assumption 1, we do not pose any requirement on the norm); and, ii) the final primal error (hence PD error) can only be worse than vanilla FW because their analysis treats $\mathbf{g}_{k+1}$ as $\nabla f(\mathbf{x}_k)$ with errors but not momentum, therefore, it is difficult to obtain the same tight PD bound as in Theorem 1. Our analytical techniques avoid these limitations.

When choosing $\delta_k = \eta_k = \frac{1}{k+1}$, we can recover Algorithm 3 in [59]. Notice that such a choice on δ_k makes $\mathbf{g}_{k+1}$ a uniform average of all gradients. A slower convergence rate $f(\mathbf{x}_k) - f(\mathbf{x}^*) = \mathcal{O}\big(\frac{LD^2 \ln k}{k}\big)$ was established in [59] through a sophisticated derivation using no-regret online learning. Through our simpler analytical framework, we can attain the same rate while providing more options for the step size.

Theorem 4. *Let Assumptions 1-3 hold, and select $\delta_k = \frac{1}{k+1}$ with η_k using one of the following options: i) $\eta_k = \frac{1}{k+1}$; ii) as in* (2.5)*; or iii) line search as in* (2.6b). *The generalized FW gap of Alg. 2 then converges with rate*

$$\mathcal{G}_k = f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k) \leq \frac{LD^2 \ln(k+1)}{2k}, \ \forall k \geq 1.$$

The rate in Theorem 4 has worse dependence on k relative to Theorems 1 and 2, partially because too much weight is put on past gradients in $\mathbf{g}_{k+1}$, suggesting that large momentum may not be helpful.

2.2.6 A side result: directional smooth step sizes

Common to both FW and HFW is that the globally estimated L might be too pessimistic for a local update. In this subsection, a local Lipschitz constant is investigated to further improve the numerical efficiency of smooth step sizes in (2.5). This easily computed local Lipschitz constant is another merit of (H)FW over projection based approaches.

[1] We are unable to derive even a primal error bound using the same analysis framework in [56] for step sizes listed in Theorem 1.

Definition 4. *(Directional Lipschitz continuous.) For two points* $\mathbf{x}, \mathbf{y} \in \mathcal{X}$*, the directional Lipschitz constant* $L(\mathbf{x}, \mathbf{y})$ *ensures* $\|\nabla f(\hat{\mathbf{x}}) - \nabla f(\hat{\mathbf{y}})\|_* \leq L(\mathbf{x}, \mathbf{y})\|\hat{\mathbf{x}} - \hat{\mathbf{y}}\|$ *for any* $\hat{\mathbf{x}} = (1-\alpha)\mathbf{x} + \alpha\mathbf{y}, \hat{\mathbf{y}} = (1-\beta)\mathbf{x} + \beta\mathbf{y}$ *with some* $\alpha \in [0,1]$ *and* $\beta \in [0,1]$.

In other words, the directional Lipschitz continuity depicts the local property on the segment between points $\mathbf{x}$ and $\mathbf{y}$. It is clear that $L(\mathbf{x}, \mathbf{y}) \leq L$. Using logistic loss for binary classification as an example, we have $L(\mathbf{x}, \mathbf{y}) \leq \frac{1}{4N}\sum_{i=1}^{N} \frac{\langle \mathbf{a}_i, \mathbf{x}-\mathbf{y}\rangle^2}{\|\mathbf{x}-\mathbf{y}\|_2^2}$, where N is the number of data, and $\mathbf{a}_i$ is the feature of the ith datum. As a comparison, the global Lipschitz constant is $L \leq \frac{1}{4N}\sum_{i=1}^{N}\|\mathbf{a}_i\|_2^2$. We show in Appendix 2.5.13 that at least for a class of functions, including widely used logistic loss and quadratic loss, $L(\mathbf{x}, \mathbf{y})$ has an analytical form.

Simply replacing L in (2.5) with $L(\mathbf{x}_k, \mathbf{v}_{k+1})$, i.e.,

$$\eta_k = \max\left\{0, \min\left\{\frac{\langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{v}_{k+1}\rangle}{L(\mathbf{x}_k, \mathbf{v}_{k+1})\|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}, 1\right\}\right\} \tag{2.7}$$

we can obtain what we term *directionally smooth step size*. Upon exploring the collinearity of $\mathbf{x}_k$, $\mathbf{x}_{k+1}$ and $\mathbf{v}_{k+1}$, a simple modification of Theorem 2 ensures the PD convergence.

Corollary 2. *Choosing* $\delta_k = \frac{2}{k+2}$*, and* η_k *via* (2.7)*, Alg. 2 ensures*

$$\mathcal{G}_k = f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k) \leq \frac{2LD^2}{k+1}, \ \forall k \geq 1.$$

The directional Lipschitz constant can also replace the global one in other FW variants, such as [45, 55], with theories therein still holding. As we shall see in numerical tests, directional smooth step sizes outperform the vanilla one by an order of magnitude.

2.3 Restart further tightens the PD error

Up till now it is established that the heavy ball momentum enables a unified analysis for tighter Type II PD bounds. In this section, we show that if the computational resources are sufficient for solving two FW subproblems per iteration, the PD error can be further improved by restart when the standard FW gap is smaller than generalized FW gap. Restart is typically employed by Nesterov's momentum in projection based methods [54] to cope with the robustness to parameter estimates, and to capture the local geometry of problem (1.1). However, it is natural to integrate restart with heavy ball momentum in FW regime. In addition, restart provides an answer to the following question: *which is smaller, the generalized FW gap or the vanilla one?* Previous works using the generalized FW gap have not addressed this question [3, 52, 58].

Algorithm 3 FW with heavy ball momentum and restart

```
1: **Initialize:** $\mathbf{x}_0^0 \in \mathcal{X}, \mathbf{g}_0^0 = \nabla f(\mathbf{x}_0^0), s \leftarrow 0, C^0 = 0, \mathcal{G}_0^0 = \bar{\mathcal{G}}_0^0$
2: **while** [not terminated] **do**
3:     $k \leftarrow 0, \mathbf{g}_0^s = \nabla f(\mathbf{x}_0^s)$
4:     **while** [$\mathcal{G}_k^s \leq \bar{\mathcal{G}}_k^s$ or $k = 0$] and [not terminated] **do**    ▷ Check whether restart is needed
5:         $\delta_k^s = \frac{2}{k+2+C^s}$
6:         $\mathbf{g}_{k+1}^s = (1 - \delta_k^s)\mathbf{g}_k^s + \delta_k^s \nabla f(\mathbf{x}_k^s)$
7:         $\mathbf{v}_{k+1}^s = \arg\min_{\mathbf{x} \in \mathcal{X}} \langle \mathbf{g}_{k+1}^s, \mathbf{x} \rangle$
8:         $\mathbf{x}_{k+1}^s = (1 - \eta_k^s)\mathbf{x}_k^s + \eta_k^s \mathbf{v}_{k+1}^s$
9:         $\bar{\mathbf{v}}_{k+1}^s = \arg\min_{\mathbf{x} \in \mathcal{X}} \langle \nabla f(\mathbf{x}_{k+1}^s), \mathbf{x} \rangle$
10:        $\mathcal{G}_{k+1}^s = f(\mathbf{x}_{k+1}^s) - \Phi_{k+1}^s(\mathbf{v}_{k+1}^s)$    ▷ Generalized FW gap
11:        $\bar{\mathcal{G}}_{k+1}^s = \langle \nabla f(\mathbf{x}_k^s), \mathbf{x}_k^s - \bar{\mathbf{v}}_{k+1}^s \rangle$    ▷ Vanilla FW gap
12:        $k \leftarrow k + 1$
13:    **end while**
14:    $K_s \leftarrow k, \mathbf{x}_0^{s+1} = \mathbf{x}_{K_s}^s, C^{s+1} = \frac{2LD^2}{\mathcal{G}_{K_s}^s}, s \leftarrow s + 1$
15: **end while**
```

FW with heavy ball momentum and restart is summarized under Alg. 3. For exposition clarity, when updating the counters such as k and s, we use notation '$\leftarrow$'. Alg. 3 contains two loops. The inner loop is the same as Alg. 2 except for computing a standard FW gap (Line 11) in addition to the generalized one (Line 10). The variable K_s, depicting the iteration number of inner loop s, is of analysis purpose. Alg. 3 can be terminated immediately whenever $\min\{\mathcal{G}_k^s, \bar{\mathcal{G}}_k^s\} \leq \epsilon$ for a desirable $\epsilon > 0$. The restart happens when the standard FW gap is smaller than generalized FW gap. And after restart, $\mathbf{g}_{k+1}^s$ will be reset. For Alg. 3, the linear functions used for generalized FW gap are defined stage-wisely

$$\Phi_0^s(\mathbf{x}) = f(\mathbf{x}_0^s) + \big\langle \nabla f(\mathbf{x}_0^s), \mathbf{x} - \mathbf{x}_0^s \big\rangle \tag{2.8a}$$

$$\Phi_{k+1}^s(\mathbf{x}) = (1 - \delta_k^s)\Phi_k^s(\mathbf{x}) + \delta_k^s \big[f(\mathbf{x}_k^s) + \big\langle \nabla f(\mathbf{x}_k^s), \mathbf{x} - \mathbf{x}_k^s \big\rangle\big], \forall\, k \geq 0. \tag{2.8b}$$

It can be verified that $\mathbf{v}_{k+1}^s$ minimizes $\Phi_{k+1}^s(\mathbf{x})$ over $\mathcal{X}$ for any $k \geq 0$. In addition, we also have $f(\mathbf{x}_0^s) - \Phi_0^s(\mathbf{v}_0^s) = \bar{\mathcal{G}}_{K_{s-1}}^{s-1}$ where $\mathbf{v}_0^s = \arg\min_{\mathbf{x} \in \mathcal{X}} \Phi_0^s(\mathbf{x})$.

There are two tunable parameters η_k^s and δ_k^s. The choice on δ_k^s has been provided directly in Line 5, where it is adaptively decided using a variable C^s relating to the generalized FW gap. Three choices are readily available for η_k^s: i) $\eta_k^s = \delta_k^s$; ii) smooth step size:

$$\eta_k^s = \max\left\{0, \min\left\{\frac{\langle \nabla f(\mathbf{x}_k^s), \mathbf{x}_k^s - \mathbf{v}_{k+1}^s \rangle}{L\|\mathbf{v}_{k+1}^s - \mathbf{x}_k^s\|^2}, 1\right\}\right\}; \tag{2.9}$$

and, iii) line search

$$\eta_k^s = \arg\min_{\eta \in [0,1]} f\big((1-\eta)\mathbf{x}_k^s + \eta \mathbf{v}_{k+1}^s\big). \tag{2.10}$$

Note that the directionally smooth step size, i.e., replacing L with $L(\mathbf{x}_k^s, \mathbf{v}_{k+1}^s)$ in (2.9) is also valid for convergence. We omit it to reduce repetition. Next we show how restart improves the PD error.

Theorem 5. *Choose η_k^s via one of the three manners: i) $\eta_k^s = \delta_k^s$; ii) as in* (2.9)*; or iii) as in* (2.10)*. If there is no restart (e.g., $s = 0$ when terminating), then Alg. 3 guarantees that*

$$\mathcal{G}_k^0 = f(\mathbf{x}_k^0) - \Phi_k(\mathbf{v}_k^0) \leq \frac{2LD^2}{k+1}, \forall k \geq 1. \tag{2.11a}$$

If restart happens, in additional to (2.11a)*, we have*

$$\mathcal{G}_k^s = f(\mathbf{x}_k^s) - \Phi_k(\mathbf{v}_k^s) < \frac{2LD^2}{k+C^s}, \forall k \geq 1, \forall s \geq 1, \;\; \textit{with} \;\; C^s \geq 1 + \sum_{j=0}^{s-1} K_j. \tag{2.11b}$$

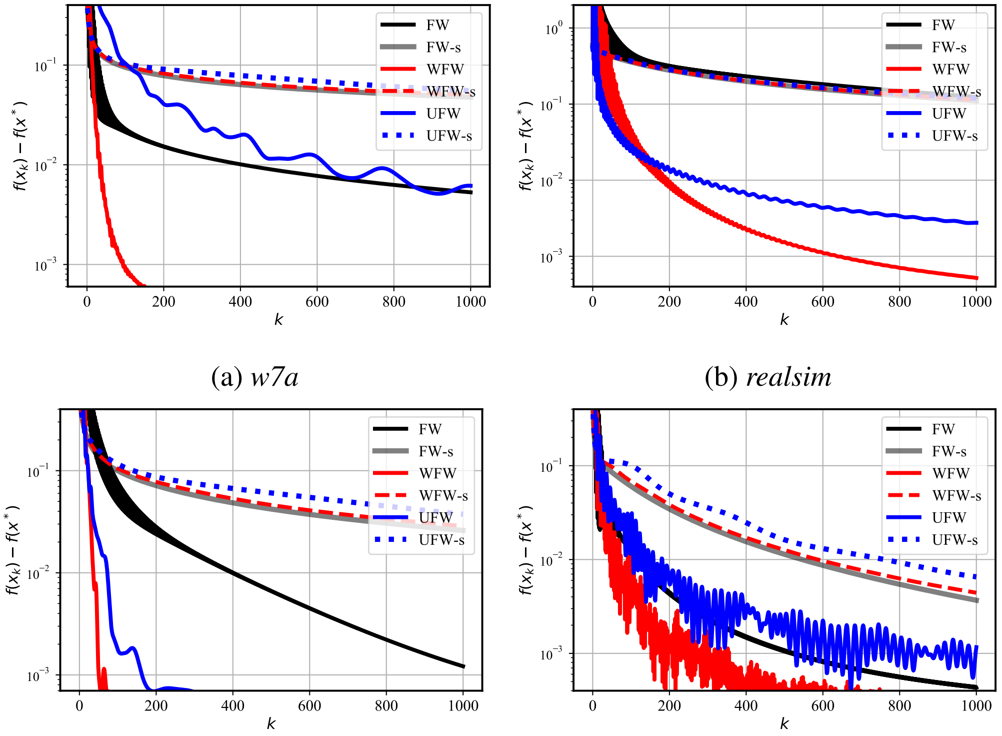
Besides the convergence of both primal and dual errors of Alg. 3, Theorem 5 implies that when no restart happens, the generalized FW gap is smaller than the standard one, demonstrating that the former is more suitable for the purpose of "stopping criterion". When restarted, Theorem 5 provides a strictly improved bound compared with Theorems 1, 2, and 3, since the denominator of the RHS in (2.11b) is no smaller than the total iteration number. An additional comparison with [52], where two subproblems are also required, once again confirms the power of heavy ball momentum to improve the constants in the PD error rate, especially with the aid of restart. The restart scheme (with slight modification) can also be employed in [52, 58, 60] to tighten their PD error.

2.4 Numerical tests

This section presents numerical tests to showcase the effectiveness of HFW on different machine learning problems. Since there are two parameters' choices for HFW in Theorems 1 and

4, we term them as weighted FW (WFW) and uniform FW (UFW), respectively, depending on the weight of $\{\nabla f(\mathbf{x}_k)\}$ in $\mathbf{g}_{k+1}$. When using smooth step size, the corresponding algorithms are marked as WFW-s and UFW-s. For comparison, the benchmark algorithms include: FW with $\eta_k = \frac{2}{k+2}$ (FW); and, FW with smooth step size (FW-s) in (1.10).

2.4.1 Binary classification



(a) *w7a*

(b) *realsim*

(c) *mushroom*

(d) *ijcnn1*

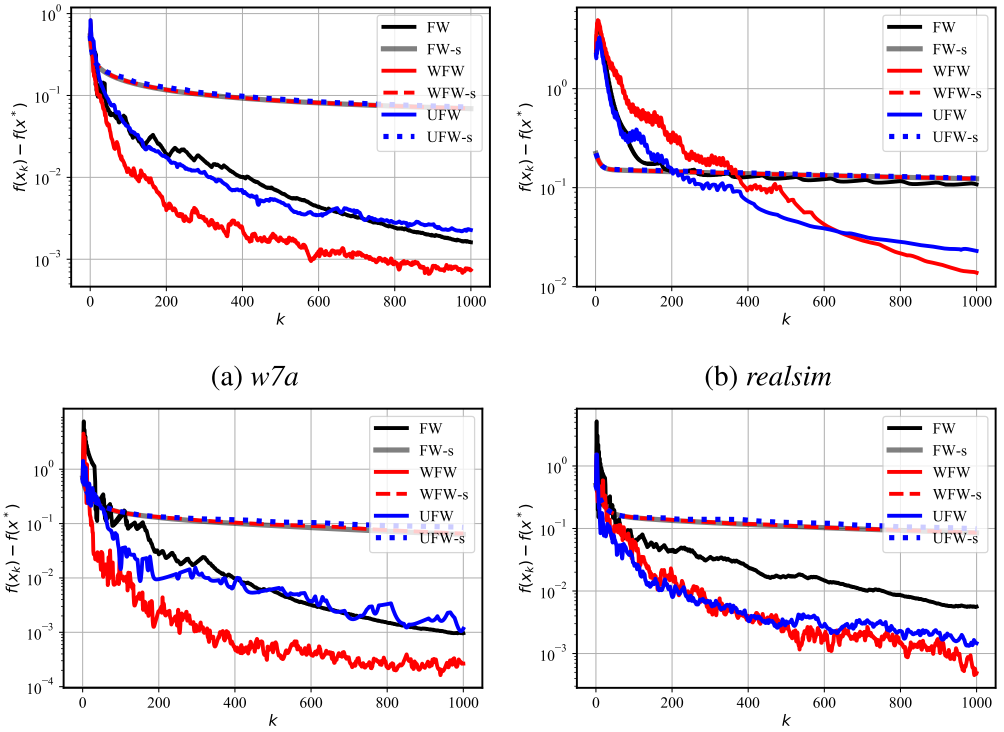
Figure 2.1: Performance of FW variants for binary classification with an ℓ_2-norm ball constraint.

We first test the performance of Alg. 2 on binary classification using logistic regression in Section 1.2.2. Datasets from LIBSVM[2] are used in the numerical tests, where details of the datasets are deferred to Appendix 2.5.14.

ℓ_2-norm ball constraint. We start with $\mathcal{X} = \{\mathbf{x} | \|\mathbf{x}\|_2 \leq R\}$. The primal errors are plotted

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

in Fig. 2.1. We use primal error here for a fair comparison. It can be seen that the parameter-free step sizes achieve better performance compared with the smooth step sizes mainly because the quality of L estimate. Such a problem can be relived through directional smooth step sizes as we shall shortly. Among parameter-free step sizes, it can be seen that WFW consistently outperforms both UFW and FW on all tested datasets, while UFW converges faster than FW only on datasets *realsim* and *mushroom*. For smooth step sizes, the per-step-descent property is validated. The excellent performance of HFW can be partially explained by the similarity of its update, namely $\mathbf{x}_{k+1} = (1-\eta_k)\mathbf{x}_k + \eta_k R \frac{\mathbf{g}_{k+1}}{\|\mathbf{g}_{k+1}\|_2}$, with normalized gradient descent (NGD) one, that is given by $\mathbf{x}_{k+1} = \text{Proj}_{\mathcal{X}}\big(\mathbf{x}_k - \eta_k \frac{\mathbf{g}_{k+1}}{\|\mathbf{g}_{k+1}\|_2}\big)$. However, there is also a subtle difference between HFW and NGD updates. Indeed, when projection is in effect, $\mathbf{x}_{k+1}$ in NGD will lie on the boundary of the ℓ_2-norm ball. Due to the convex combination nature of the update in HFW, it is unlikely to have $\mathbf{x}_{k+1}$ on the boundary, though it can come arbitrarily close.



(a) *w7a*

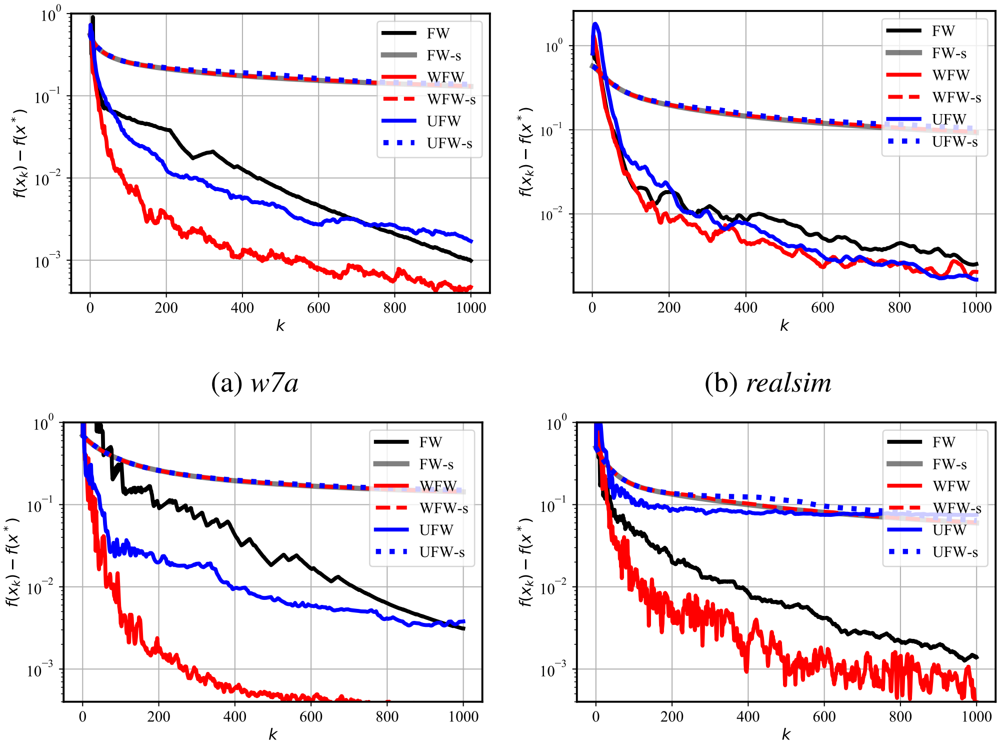
(b) *realsim*

(c) *mushroom*

(d) *ijcnn1*

Figure 2.2: Performance of FW variants for binary classification with an ℓ_1-norm ball constraint.

ℓ_1**-norm ball constraint.** Here $\mathcal{X} = \{\mathbf{x} | \|\mathbf{x}\|_1 \leq R\}$ denotes the constraint set that promotes sparse solutions. In the simulation, R is tuned for a solution with similar sparsity as the dataset itself. The results are showcased in Fig. 2.2. For smooth step sizes, FW-s, UFW-s, and WFW-s exhibit similar performances, and their curves are smooth. On the other hand, parameter-free step sizes eventually outperform smooth step sizes though the curves zig-zag. (The curves on *realsim* are smoothed to improve figure quality.) UFW has similar performance on *w7a* and *mushroom* with FW and faster convergence on other datasets. Once again, WFW consistently outperforms FW and UFW.



(a) *w7a*

(b) *realsim*

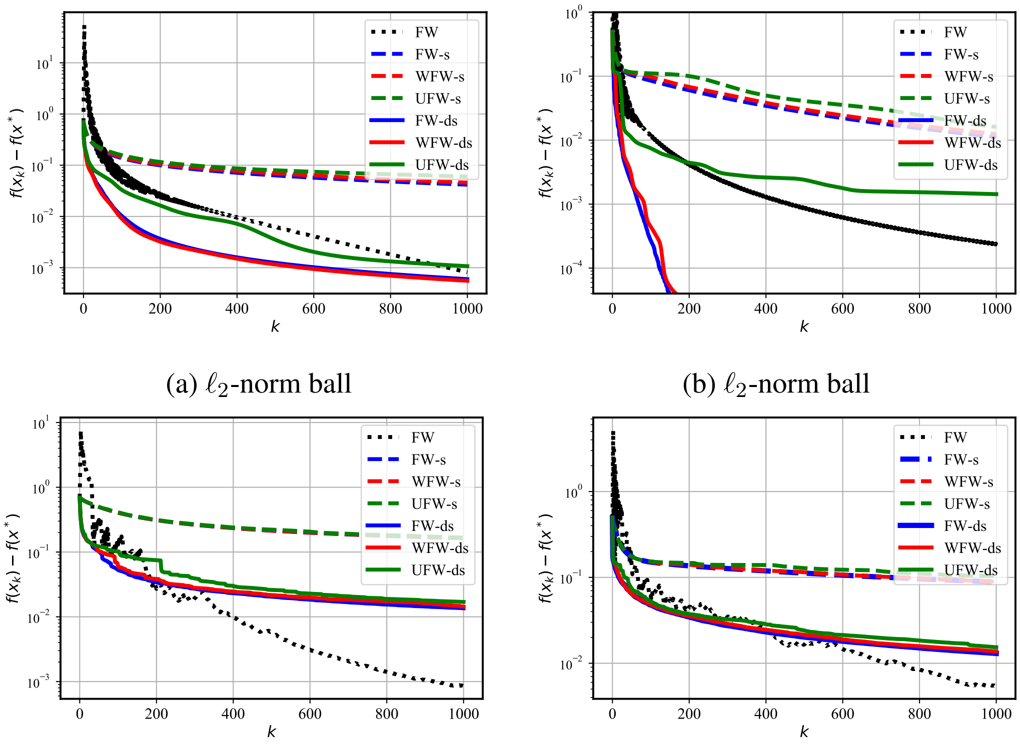
(c) *mushroom*

(d) *ijcnn1*

Figure 2.3: Performance of FW variants for binary classification with an n-support norm ball constraint.

n**-support norm ball constraint.** The n-support norm ball is a tighter relaxation of a sparsity enforcing ℓ_0-norm ball combined with an ℓ_2-norm penalty compared with ElasticNet [61]. It gives rise to $\mathcal{X} = \text{conv}\{\mathbf{x} | \|\mathbf{x}\|_0 \leq n, \|\mathbf{x}\|_2 \leq R\}$, where $\text{conv}\{\cdot\}$ denotes the convex hull [12]. The closed-form solution of $\mathbf{v}_{k+1}$ is given in [62]. In the simulation, we choose

$n = 2$ and tune R for a solution whose sparsity is similar to the adopted dataset. The results are showcased in Fig. 2.3. For smooth step sizes, FW-s and WFW-s exhibit similar performance, while UFW-s converges slightly slower on *ijcnn1*. Regarding parameter-free step sizes, UFW does not offer faster convergence compared with FW on the tested datasets, but WFW again has numerical merits.



(a) ℓ_2-norm ball

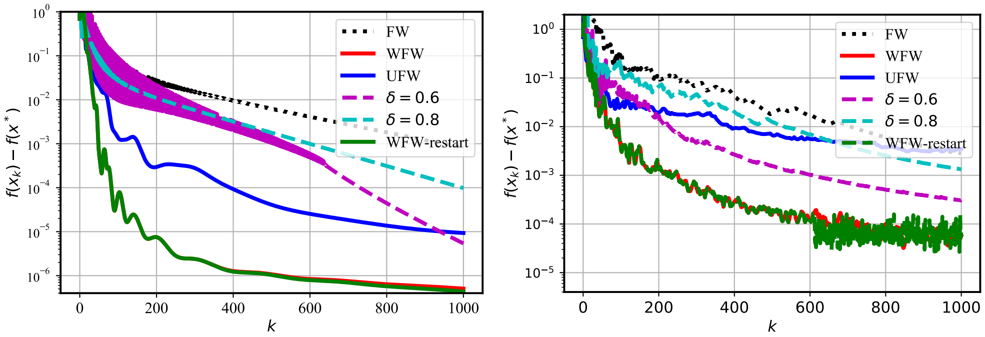
(b) ℓ_2-norm ball

(c) ℓ_1-norm ball

(d) ℓ_1-norm ball

Figure 2.4: Performance of directionally smooth step sizes. (a) and (c) are tested on *mushroom*; and (b) and (d) use *ijcnn1*.

Directionally smooth step sizes. The results in Fig. 2.4 validate the effectiveness of directionally smooth (-ds) step sizes. For all datasets tested, the benefit of adopting $L(\mathbf{x}_k, \mathbf{v}_{k+1})$ is evident, as it improves the performance of smooth step sizes by an order of magnitude. In addition, it is also observed that UFW-ds performs worse than WFW-ds, which suggests that putting too much weight on past gradients could be less attractive in practice.



(a) ℓ_2 norm ball

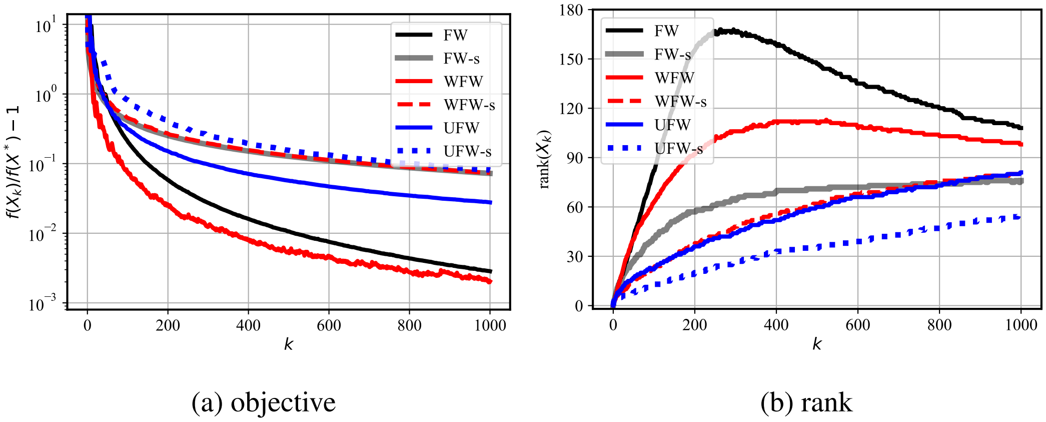
(b) n-supp norm ball

Figure 2.5: Comparison of HFW with other algorithms on muchroom.

Additional comparisons. We also compare HFW with a generalized version of [56], where we set $\delta_k = \delta \in (0, 1), \forall k$ in Alg. 2. Two specific choices, i.e., $\delta = 0.6$, and $\delta = 0.8$, are plotted in Fig. 2.5, where the ℓ_2-norm ball and n-support norm ball are adopted as constraints. In both cases, WFW converges faster than the algorithm adapted from [56, 63]. In addition, the choice of δ has major impact on convergence behavior, while WFW avoids this need for manual tuning of δ. The performance of WFW with restart, i.e., Alg. 3, is also shown in Fig. 2.5. Although it slightly outperforms WFW, restart also doubles the computational burden due to the need of solving two FW subproblems. From this point of view, WFW with restart is more of theoretical rather than practical interest. In addition, it is observed that Alg. 3 is not restarted after the first few iterations, which suggests that the generalized FW gap is smaller than the vanilla one, at least in the early stage of convergence. Thus, the generalized FW gap is attractive as a stopping criterion when a solution with moderate accuracy is desirable.

In a nutshell, the numerical experiments suggest that heavy ball momentum performs best with parameter-free step sizes with the momentum weight carefully adjusted. WFW is mainly recommended because it achieves improved empirical performance compared to UFW and FW, regardless of the constraint sets. The smooth step sizes on the other hand, eliminate the zig-zag behavior at the price of convergence slowdown due to the need of L, while directionally smooth step sizes can be helpful to alleviate this convergence slowdown.

2.4.2 Matrix completion



(a) objective

(b) rank

Figure 2.6: Performance of FW variants for matrix completion on *MovieLens100K*.

This subsection focuses on matrix completion problems for recommender systems. The problem to be solved can be found in Section 1.2.3.

Heavy ball based FW are tested using dataset *MovieLens100K*[3] . Following the initialization of [18], the numerical results can be found in Fig. 2.6. Subfigures (a) and (b) depict the optimality error and rank versus k for $R = 3$. For parameter-free step sizes, WFW converges faster than FW while finding solutions with lower rank. The low rank solution of UFW is partially because it does not converge sufficiently. For smooth step sizes, UFW-s finds a solution with slightly larger objective value but much lower rank compared with WFW-s and FW-s. Overall, when a small optimality error is the priority, WFW is more attractive; while UFW-s is useful for finding low rank solutions.

2.5 Appendix

2.5.1 $f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k)$ for the smooth step sizes in Alg. 2

When using the step size (2.7) in Alg. 2, $f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k)$ is ensured.

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) \leq \eta_k \langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \rangle + \frac{\eta_k^2 L}{2} \|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2 \leq 0$$

[3] `https://grouplens.org/datasets/movielens/100k/`

where the last ineqaulity is because η_k minimizes $\eta\langle\nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k\rangle + \frac{\eta^2 L}{2}\|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2$ over $[0, 1]$.

2.5.2 Proof of Lemma 1

Proof. Using $\mathbf{g}_{k+1} = \sum_{\tau=0}^{k} w_k^\tau \nabla f(\mathbf{x}_\tau)$, we have

$$\arg\min_{\mathbf{x}\in\mathcal{X}} \Phi_{k+1}(\mathbf{x}) = \arg\min_{\mathbf{x}\in\mathcal{X}} \Big\langle \sum_{\tau=0}^{k} w_k^\tau \nabla f(\mathbf{x}_\tau), \mathbf{x} \Big\rangle = \arg\min_{\mathbf{x}\in\mathcal{X}} \big\langle \mathbf{g}_{k+1}, \mathbf{x} \big\rangle.$$

By comparing with Line 4 of Alg. 2, one can see that $\mathbf{v}_{k+1}$ is a minimizer of $\Phi_{k+1}(\mathbf{x})$ over $\mathcal{X}$. To prove that $\Phi_{k+1}(\mathbf{x})$ is a lower bound of $f(\mathbf{x})$, we appeal to convexity to write

$$\Phi_{k+1}(\mathbf{x}) = \sum_{\tau=0}^{k} w_k^\tau \big[f(\mathbf{x}_\tau) + \langle \nabla f(\mathbf{x}_\tau), \mathbf{x} - \mathbf{x}_\tau \rangle \big] \leq \sum_{\tau=0}^{k} w_k^\tau f(\mathbf{x}) = f(\mathbf{x})$$

where the last equation is because $\sum_{\tau=0}^{k} w_k^\tau = 1$ holds for any k. The proof is thus complete. □

2.5.3 Proof of Theorem 1

Proof. Using Assumption 1, we have

$$\begin{aligned} & f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) \\ & \leq \big\langle \nabla f(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k \big\rangle + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\ & = \eta_k \big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \big\rangle + \frac{\eta_k^2 L}{2}\|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2. \end{aligned} \tag{2.12}$$

Inequality (2.12) is standard in the analysis of FW and its variants. Letting $\Phi_0(\mathbf{x}) \equiv 0$, and $\mathbf{v}_0$ be any point in $\mathcal{X}$, it can be verified that $\Phi_{k+1}(\mathbf{x}) = (1 - \delta_k)\Phi_k(\mathbf{x}) + \delta_k \big[f(\mathbf{x}_k) + \big\langle \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \big\rangle \big]$, from which we have

$$\begin{aligned} & \Phi_{k+1}(\mathbf{v}_{k+1}) \\ & = (1 - \delta_k)\Phi_k(\mathbf{v}_{k+1}) + \delta_k \Big[f(\mathbf{x}_k) + \big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \big\rangle \Big] \\ & \overset{(a)}{\geq} (1 - \delta_k)\Phi_k(\mathbf{v}_k) + \delta_k \Big[f(\mathbf{x}_k) + \big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \big\rangle \Big] \end{aligned} \tag{2.13}$$

where (a) is because $1 - \delta_k \geq 0$ and $\mathbf{v}_k$ minimizes $\Phi_k(\mathbf{x})$ over $\mathcal{X}$ (hence $\Phi_k(\mathbf{v}_k) \leq \Phi_k(\mathbf{v}_{k+1})$). Now subtracting $\Phi_{k+1}(\mathbf{v}_{k+1})$ on both sides of (2.12), we have

$$\begin{aligned} & f(\mathbf{x}_{k+1}) - \Phi_{k+1}(\mathbf{v}_{k+1}) \\ \overset{(b)}{\leq} \; & (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + \frac{\delta_k^2 L \|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2} \\ \overset{(c)}{\leq} \; & (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + \frac{\delta_k^2 L D^2}{2} \end{aligned} \tag{2.14}$$

where (b) uses $\eta_k = \delta_k$ and (2.13); and (c) relies on Assumption 3. For convenience, let $\Delta(i,j) := \prod_{\tau=i}^{j}(1-\delta_\tau)$, and unroll (2.14) to arrive at

$$\begin{aligned} & f(\mathbf{x}_{k+1}) - \Phi_{k+1}(\mathbf{v}_{k+1}) \\ \leq \; & \Delta(0,k)\big[f(\mathbf{x}_0) - \Phi_0(\mathbf{v}_0)\big] + \sum_{\tau=0}^{k} \frac{LD^2\delta_\tau^2}{2}\Delta(\tau+1,k). \end{aligned}$$

Plugging in the values of δ_k completes the proof. □

2.5.4 Proof of Theorem 2

Proof. The first a few steps are the same as the proof of Theorem 1; i.e., we have (2.12) and (2.13). Combining (2.12) and (2.13), we arrive at

$$\begin{aligned} & f(\mathbf{x}_{k+1}) - \Phi_{k+1}(\mathbf{v}_{k+1}) \\ \leq \; & (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + (\eta_k - \delta_k)\big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \big\rangle + \frac{\eta_k^2 L \|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2}. \end{aligned} \tag{2.15}$$

It can be verified that the specific choice of η_k minimizes the RHS of (2.15) over $[0, 1]$. Hence we have

$$\begin{aligned}
& f(\mathbf{x}_{k+1}) - \Phi_{k+1}(\mathbf{v}_{k+1}) \qquad (2.16)\\
&\leq (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + \frac{\eta_k^2 L\|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2} + (\eta_k - \delta_k)\big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \big\rangle\\
&\overset{(a)}{\leq} (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + \frac{\alpha_k^2 L\|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2} + (\alpha_k - \delta_k)\big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \big\rangle\\
&\overset{(b)}{=} (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + \frac{\delta_k^2 L\|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2}\\
&\leq \big[f(\mathbf{x}_0) - \Phi_0(\mathbf{v}_0)\big]\prod_{\tau=0}^{k}(1-\delta_\tau) + \sum_{\tau=0}^{k}\frac{LD^2\delta_\tau^2}{2}\prod_{j=\tau+1}^{k}(1-\delta_j)\\
&\leq \frac{2LD^2}{k+2}
\end{aligned}$$

where in (a) α_k can be chosen as any number in $[0, 1]$; in (b) we set $\alpha_k = \delta_k$. This completes the proof. □

2.5.5 An extension of Theorem 2 for per step descent of $\mathcal{G}_k$

In this section, we show that it is possible to ensure per step descent on generalized FW gap when a more difficult subproblem can be solved. In particular, we will replace Line 4 of Alg. 2 and choose parameters as

$$(\delta_k, \mathbf{v}_{k+1}) = \underset{\delta\in[0,1], \mathbf{v}\in\mathcal{X}}{\arg\min}\ (1-\delta)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + \frac{\delta^2 L\|\mathbf{v} - \mathbf{x}_k\|^2}{2} \qquad (2.17a)$$

$$\eta_k = \delta_k. \qquad (2.17b)$$

It is clear that (2.17a) is harder to solve compared with a FW subproblem. The choice of δ_k enables an adaptive weights for $\nabla f(\mathbf{x}_k)$ in $\mathbf{g}_{k+1}$. Next we present the main result for such a parameter choice.

Theorem 6. *When Assumptions 1, 2 and 3 are satisfied, choosing* $\mathbf{v}_{k+1}$*,* η_k *and* δ_k *according to* (2.17)*, Alg. 2 guarantees that: i)* $\mathcal{G}_{k+1} \leq \mathcal{G}_k$*, and ii)*

$$\mathcal{G}_k = f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k) \leq \frac{2LD^2}{k+1}, \ \forall k \geq 1.$$

Proof. It can be seen that (2.15) still holds, from which we have

$$\begin{aligned} & f(\mathbf{x}_{k+1}) - \Phi_{k+1}(\mathbf{v}_{k+1}) && (2.18) \\ \leq\, & (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + \frac{\eta_k^2 L \|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2} + (\eta_k - \delta_k)\big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \big\rangle \\ \overset{(a)}{=}\, & (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + \frac{\delta_k^2 L \|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2} \end{aligned}$$

where (a) is because $\eta_k = \delta_k$. Then by the manner δ_k is chosen, we have

$$\begin{aligned} & f(\mathbf{x}_{k+1}) - \Phi_{k+1}(\mathbf{v}_{k+1}) && (2.19) \\ =\, & (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + \frac{\delta_k^2 L \|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2} \\ \overset{(b)}{\leq}\, & (1-\tilde{\delta}_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + \frac{\tilde{\delta}_k^2 L \|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2} \end{aligned}$$

where in (b) $\tilde{\delta}_k \in [0,1]$. Choosing $\tilde{\delta}_k = 0$, we obtain $\mathcal{G}_{k+1} \leq \mathcal{G}_k$. Choosing $\tilde{\delta}_k = \frac{2}{k+2}$, we obtain the convergence rate. □

2.5.6 Line search for Alg. 2

Proof. Let $\tilde{\eta}_k = \frac{2}{k+2}, \forall k$. By the choice of η_k, we have

$$f(\mathbf{x}_{k+1}) = \min_{\eta \in [0,1]} f\big((1-\eta)\mathbf{x}_k + \eta \mathbf{v}_{k+1}\big) \leq f\big((1-\tilde{\eta}_k)\mathbf{x}_k + \tilde{\eta}_k \mathbf{v}_{k+1}\big). \qquad (2.20)$$

Then using smoothness, we arrive at

$$\begin{aligned} & f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) && (2.21) \\ \leq\, & f\big((1-\tilde{\eta}_k)\mathbf{x}_k + \tilde{\eta}_k \mathbf{v}_{k+1}\big) - f(\mathbf{x}_k) \\ \leq\, & \tilde{\eta}_k \big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \big\rangle + \frac{\tilde{\eta}_k^2 L}{2} \|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2. \end{aligned}$$

Then combining (2.21) and (2.13), and following the same steps in (2.14), we can prove this theorem. □

Through Theorem 3 it is straightforward to derive the primal and dual convergence, respectively, following the same argument of Corollary 1. For this reason, it is omitted here.

2.5.7 Proof of Theorem 4

Proof. It can be seen that (2.15) still holds.

Parameter-free step size. Plugging in $\delta_k = \eta_k = \frac{1}{k+1}$ into (2.15), we arrive at

$$\begin{aligned} f(\mathbf{x}_{k+1}) - \Phi_{k+1}(\mathbf{v}_{k+1}) &\leq (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + \frac{\delta_k^2 L \|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2} \\ &\leq \Delta(0,k)\big[f(\mathbf{x}_0) - \Phi_0(\mathbf{v}_0)\big] + \sum_{\tau=0}^{k} \frac{LD^2\delta_\tau^2}{2}\Delta(\tau+1,k) \\ &= \mathcal{O}\Big(\frac{LD^2 \ln(k+2)}{k+1}\Big) \end{aligned} \tag{2.22}$$

where $\Delta(i,j) := \prod_{\tau=i}^{j}(1-\delta_\tau)$, $\Phi_0(\mathbf{x}) \equiv 0$, and $\mathbf{v}_0$ is any point in $\mathcal{X}$.

Smooth step size. Notice that the choice of η_k minimizes the RHS of (2.15) when δ_k is fixed, then we have

$$\begin{aligned} & f(\mathbf{x}_{k+1}) - \Phi_{k+1}(\mathbf{v}_{k+1}) \\ & \leq (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + (\eta_k - \delta_k)\big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \big\rangle + \frac{\eta_k^2 L \|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2} \\ & \overset{(a)}{\leq} (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + (\tilde{\eta}_k - \delta_k)\big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \big\rangle + \frac{\tilde{\eta}_k^2 L \|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2} \\ & \overset{(b)}{\leq} (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + \frac{\delta_k^2 L \|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2} \\ & = \mathcal{O}\Big(\frac{LD^2 \ln(k+2)}{k+1}\Big) \end{aligned} \tag{2.23}$$

where in (a) $\tilde{\eta}_k \in [0,1]$; and in (b) we set $\tilde{\eta}_k = \delta_k$.

Line search. When η_k is chosen via line search, we have for any $\tilde{\eta}_k \in [0,1]$

$$f(\mathbf{x}_{k+1}) = \min_{\eta \in [0,1]} f\big((1-\eta)\mathbf{x}_k + \eta \mathbf{v}_{k+1}\big) \leq f\big((1-\tilde{\eta}_k)\mathbf{x}_k + \tilde{\eta}_k \mathbf{v}_{k+1}\big). \tag{2.24}$$

Then by smoothness, we have

$$\begin{aligned} f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) &\leq f\big((1-\tilde{\eta}_k)\mathbf{x}_k + \tilde{\eta}_k \mathbf{v}_{k+1}\big) - f(\mathbf{x}_k) \\ &\leq \tilde{\eta}_k \big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \big\rangle + \frac{\tilde{\eta}_k^2 L}{2}\|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2. \end{aligned} \tag{2.25}$$

Then using the same argument as the derivation of (2.15), we can obtain

$$\begin{aligned} & f(\mathbf{x}_{k+1}) - \Phi_{k+1}(\mathbf{v}_{k+1}) \\ & \leq (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + (\tilde{\eta}_k - \delta_k)\big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \big\rangle + \frac{\tilde{\eta}_k^2 L \|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2}. \end{aligned} \tag{2.26}$$

Simply setting $\tilde{\eta}_k = \frac{1}{k+1}$, and using the same derivation as in (2.23), the proof can be completed.

□

2.5.8 Proof for choosing $\delta_k = \delta$

When Assumptions 1 is satisfied w.r.t. ℓ_2-norm, we show the following parameter choice in Alg. 2 leads to convergence as well.

$$\delta_k = \delta, \ \eta_k = \frac{c}{k+k_0}, \ \forall k \geq 0 \tag{2.27}$$

where $\delta \in (0,1)$, and c and k_0 are constants to be specified later. Due to the choice of $\delta_k = \delta$, $\mathbf{g}_{k+1}$ is an exponentially moving average of previous gradients. Note that the moving average was adopted in [56] for stochastic FW to reduce the mean square error of the noisy gradient. However, we use it in a totally different purpose.

Lemma 2. *Choose parameters as in* (2.27)*. Suppose there exist a constant* c_0 *that satisfies*

$$c_1^2 \leq \left[1 - (1-\delta)\frac{(k_0+1)^2}{k_0^2}\right]\delta c_0^2 \tag{2.28}$$

then it is guaranteed that

$$\|\mathbf{g}_{k+1} - \nabla f(\mathbf{x}_k)\|_2^2 \leq \frac{c_0^2 L^2 D^2}{(k+k_0)^2}.$$

Proof.

$$\begin{aligned}
&\|\mathbf{g}_{k+1} - \nabla f(\mathbf{x}_k)\|_2^2 \\
&= (1-\delta)^2 \|\mathbf{g}_k - \nabla f(\mathbf{x}_k)\|_2^2 \\
&= (1-\delta)^2 \|\mathbf{g}_k - \nabla f(\mathbf{x}_{k-1}) + \nabla f(\mathbf{x}_{k-1}) - \nabla f(\mathbf{x}_k)\|_2^2 \\
&\overset{(a)}{\leq} (1-\delta)^2(1+\theta)\|\mathbf{g}_k - \nabla f(\mathbf{x}_{k-1})\|_2^2 + (1-\delta)^2(1+\frac{1}{\theta})\|\nabla f(\mathbf{x}_{k-1}) - \nabla f(\mathbf{x}_k)\|_2^2 \\
&\overset{(b)}{\leq} (1-\delta)^2(1+\theta)\|\mathbf{g}_k - \nabla f(\mathbf{x}_{k-1})\|_2^2 + (1-\delta)^2(1+\frac{1}{\theta})L^2\eta_{k-1}^2\|\mathbf{x}_{k-1} - \mathbf{v}_k\|_2^2 \\
&\overset{(c)}{\leq} (1-\delta)^2(1+\theta)\|\mathbf{g}_k - \nabla f(\mathbf{x}_{k-1})\|_2^2 + (1-\delta)^2(1+\frac{1}{\theta})L^2D^2\eta_{k-1}^2 \\
&\overset{(d)}{\leq} (1-\delta)\|\mathbf{g}_k - \nabla f(\mathbf{x}_{k-1})\|_2^2 + (1-\delta)^2(1+\frac{1}{\delta})L^2D^2\eta_{k-1}^2 \\
&\overset{(e)}{\leq} (1-\delta)\|\mathbf{g}_k - \nabla f(\mathbf{x}_{k-1})\|_2^2 + L^2D^2\frac{\eta_{k-1}^2}{\delta}
\end{aligned} \tag{2.29}$$

where (a) is by Young's inequality with $\theta > 0$ to be specified later; (b) follows from Assumption 1; (c) is because Assumption 3; in (d) we choose $\theta = \delta < 1$ and use the fact that $(1-\delta)^2(1+\delta) \leq (1-\delta)$; and (e) uses $\delta \leq 1$ so that $(1-\delta)^2(1+\frac{1}{\delta}) = \frac{1}{\delta} - 1 + \delta^2 - 2\delta \leq \frac{1}{\delta}$.

We proof this lemma by induction. Given the choice of $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$, we must have $\mathbf{g}_1 = \nabla f(\mathbf{x}_0)$, which implies $\|\mathbf{g}_1 - \nabla f(\mathbf{x}_0)\|_2^2 = 0 \leq \frac{c_0^2 L^2 D^2}{k_0^2}$ directly. Next we assume that $\|\mathbf{g}_k - \nabla f(\mathbf{x}_{k-1})\|_2^2 \leq \frac{c_0^2 L^2 D^2}{(k-1+k_0)^2}$ holds for some $k \geq 1$. Using (2.29), we have

$$\begin{aligned}
\|\mathbf{g}_{k+1} - \nabla f(\mathbf{x}_k)\|_2^2 &\leq (1-\delta)\|\mathbf{g}_k - \nabla f(\mathbf{x}_{k-1})\|_2^2 + L^2 D^2 \frac{\eta_{k-1}^2}{\delta} \\
&\leq (1-\delta)\frac{c_0^2 L^2 D^2}{(k+k_0-1)^2} + L^2 D^2 \frac{\eta_{k-1}^2}{\delta} \\
&\leq (1-\delta)\frac{c_0^2 L^2 D^2}{(k+k_0-1)^2} + L^2 D^2 \frac{c_1^2}{\delta(k+k_0)^2} \\
&= (1-\delta)\frac{c_0^2 L^2 D^2}{(k+k_0)^2}\frac{(k+k_0)^2}{(k+k_0-1)^2} + L^2 D^2 \frac{c_1^2}{\delta(k+k_0)^2} \\
&\leq (1-\delta)\frac{c_0^2 L^2 D^2}{(k+k_0)^2}\frac{(k_0+1)^2}{k_0^2} + L^2 D^2 \frac{c_1^2}{\delta(k+k_0)^2} \\
&\leq \frac{c_0^2 L^2 D^2}{(k+k_0)^2} \qquad (2.30)
\end{aligned}$$

where the last inequality comes from the choice of c_1. The proof is thus completed. □

To avoid the complexity of choosing constants, we consider an instance where $k_0 = 2$, $\delta = 0.8$, $c_1 = 2$, and $c_0 \approx 3.05$. It can be verified that (2.28) is satisfied. Then applying Lemma 2, the convergence of Alg.2 can be obtained.

Theorem 7. *Let* $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$, $\eta_k = \frac{2}{k+3}$, *and* $\delta = 0.8$. *Then for* $\forall k \geq 1$, *the convergence rate of Alg. 2 with* (2.27) *is*

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) = \mathcal{O}\Big(\frac{LD^2}{k}\Big).$$

Proof. Using Assumption 1, we have

$$\begin{aligned}
f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*) &\leq f(\mathbf{x}_k) - f(\mathbf{x}^*) + \big\langle \nabla f(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k \big\rangle + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2^2 \qquad (2.31) \\
&= f(\mathbf{x}_k) - f(\mathbf{x}^*) + \eta_k \big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \big\rangle + \frac{\eta_k^2 L}{2}\|\mathbf{v}_{k+1} - \mathbf{x}_k\|_2^2 \\
&\leq f(\mathbf{x}_k) - f(\mathbf{x}^*) + \eta_k \big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \big\rangle + \frac{\eta_k^2 L D^2}{2}.
\end{aligned}$$

Next we have

$$\begin{aligned}\big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \big\rangle &= \big\langle \nabla f(\mathbf{x}_k), \mathbf{x}^* - \mathbf{x}_k \big\rangle + \big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}^* \big\rangle \\ &\overset{(a)}{\leq} f(\mathbf{x}^*) - f(\mathbf{x}_k) + \big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}^* \big\rangle \\ &= f(\mathbf{x}^*) - f(\mathbf{x}_k) + \big\langle \mathbf{g}_{k+1}, \mathbf{v}_{k+1} - \mathbf{x}^* \big\rangle + \big\langle \nabla f(\mathbf{x}_k) - \mathbf{g}_{k+1}, \mathbf{v}_{k+1} - \mathbf{x}^* \big\rangle \\ &\overset{(b)}{\leq} f(\mathbf{x}^*) - f(\mathbf{x}_k) + \big\langle \nabla f(\mathbf{x}_k) - \mathbf{g}_{k+1}, \mathbf{v}_{k+1} - \mathbf{x}^* \big\rangle \\ &\leq f(\mathbf{x}^*) - f(\mathbf{x}_k) + D \|\nabla f(\mathbf{x}_k) - \mathbf{g}_{k+1}\|_2 \qquad (2.32)\end{aligned}$$

where (a) is by the convexity of $f(\mathbf{x})$; (b) is because $\mathbf{v}_{k+1}$ minimizes $\langle \mathbf{g}_{k+1}, \mathbf{x}\rangle$ over $\mathcal{X}$; and the last inequality relies on Cauchy-Schwarz inequality and Assumption 3. Plugging (2.32) into (2.31), we have

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*) \leq (1-\eta_k)\big[f(\mathbf{x}_k) - f(\mathbf{x}^*)\big] + \eta_k D \|\nabla f(\mathbf{x}_k) - \mathbf{g}_{k+1}\|_2 + \frac{\eta_k^2 L D^2}{2}. \qquad (2.33)$$

Let $\xi_k = \frac{\eta_k c_0 L D^2}{k+k_0} + \frac{\eta_k^2 L D^2}{2}$, then we have

$$\begin{aligned}f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*) &\leq (1-\eta_k)\big[f(\mathbf{x}_k) - f(\mathbf{x}^*)\big] + \eta_k D \|\nabla f(\mathbf{x}_k) - \mathbf{g}_{k+1}\|_2 + \frac{\eta_k^2 L D^2}{2} \\ &\leq (1-\eta_k)\big[f(\mathbf{x}_k) - f(\mathbf{x}^*)\big] + \xi_k \\ &= \big[f(\mathbf{x}_0) - f(\mathbf{x}^*)\big] \prod_{\tau=0}^{k} (1-\eta_\tau) + \sum_{\tau=0}^{k} \xi_\tau \prod_{j=\tau+1}^{k} (1-\eta_j) \\ &= \mathcal{O}\Big(\frac{LD^2}{k}\Big). \qquad (2.34)\end{aligned}$$

The proof is thus completed. □

2.5.9 Additional discussions

Many of existing works, e.g., [47], study (projected) heavy ball momentum by introducing auxiliary variables $\mathbf{z}_k$ such that the update on variable $\mathbf{x}_k$ can be viewed as a "gradient update" on $\mathbf{z}_k$, i.e., $\mathbf{z}_{k+1} = \mathbf{z}_k - \eta \nabla f(\mathbf{x}_k)$. By constructing the $\{\mathbf{z}_k\}$ sequence, it is possible to view heavy ball momentum approximately as GD. Though this trick is smart and analytically convenient, it does not give too much insight for the heavy ball momentum itself.

By comparing the use of heavy ball momentum in FW and GD, it may suggest new perspectives. For example, one can view Alg.2 as the dual-averaging version of FW as well. This suggests that it is intriguing to study (projected) heavy ball momentum from dual-averaging point of view. This is slightly off the main theme of this thesis, and we leave it for future research.

2.5.10 Stopping criterion

Recall that for a prescribed $\epsilon > 0$, having $f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k) \leq \epsilon$ directly implies $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \epsilon$. Next, we show how to update $\Phi_k(\mathbf{v}_k)$ iteratively in order to obtain a stopping criterion. Let us note that

$$\begin{aligned}\Phi_{k+1}(\mathbf{x}) &= \sum_{\tau=0}^{k} w_k^\tau \big[f(\mathbf{x}_\tau) + \langle \nabla f(\mathbf{x}_\tau), \mathbf{x} - \mathbf{x}_\tau \rangle\big] \\ &= \sum_{\tau=0}^{k} w_k^\tau \big[f(\mathbf{x}_\tau) - \langle \nabla f(\mathbf{x}_\tau), \mathbf{x}_\tau \rangle\big] + \langle \mathbf{g}_{k+1}, \mathbf{x} \rangle \\ &:= C_{k+1} + \langle \mathbf{g}_{k+1}, \mathbf{x} \rangle, \ \forall k \geq 0.\end{aligned}$$

Hence, to compute $\Phi_{k+1}(\mathbf{v}_{k+1})$, we only need to update C_{k+1} iteratively. A simple derivation leads to

$$\begin{aligned}C_{k+1} = (1-\delta_k)C_k + \delta_k \Big[f(\mathbf{x}_k) - \langle \nabla f(\mathbf{x}_k), \mathbf{x}_k \rangle \Big], \\ \text{with } \ C_1 = f(\mathbf{x}_0) - \langle \nabla f(\mathbf{x}_0), \mathbf{x}_0 \rangle.\end{aligned} \tag{2.35}$$

In sum, one can efficiently obtain $\Phi_{k+1}(\mathbf{v}_{k+1})$ as

$$\Phi_{k+1}(\mathbf{v}_{k+1}) = C_{k+1} + \langle \mathbf{g}_{k+1}, \mathbf{v}_{k+1} \rangle \tag{2.36}$$

with C_{k+1} recursively updated via (2.35).

2.5.11 Proof of Theorem 5

Proof. Consider the case where $\eta_k^s = \delta_k^s$. Using Assumption 1, we have

$$\begin{aligned}f(\mathbf{x}_{k+1}^s) - f(\mathbf{x}_k^s) &\leq \big\langle \nabla f(\mathbf{x}_k^s), \mathbf{x}_{k+1}^s - \mathbf{x}_k^s \big\rangle + \frac{L}{2}\|\mathbf{x}_{k+1}^s - \mathbf{x}_k^s\|^2 \qquad (2.37) \\ &= \eta_k^s \big\langle \nabla f(\mathbf{x}_k^s), \mathbf{v}_{k+1}^s - \mathbf{x}_k^s \big\rangle + \frac{(\eta_k^s)^2 L}{2}\|\mathbf{v}_{k+1}^s - \mathbf{x}_k^s\|^2.\end{aligned}$$

Then we have

$$\begin{aligned}\Phi_{k+1}^s(\mathbf{v}_{k+1}^s) &= (1-\delta_k^s)\Phi_k^s(\mathbf{v}_{k+1}^s) + \delta_k^s\Big[f(\mathbf{x}_k^s) + \big\langle \nabla f(\mathbf{x}_k^s), \mathbf{v}_{k+1}^s - \mathbf{x}_k^s\big\rangle\Big] \\ &\geq (1-\delta_k^s)\Phi_k^s(\mathbf{v}_k^s) + \delta_k^s\Big[f(\mathbf{x}_k^s) + \big\langle \nabla f(\mathbf{x}_k^s), \mathbf{v}_{k+1}^s - \mathbf{x}_k^s\big\rangle\Big].\end{aligned} \tag{2.38}$$

Now subtracting $\Phi_{k+1}^s(\mathbf{v}_{k+1}^s)$ on both sides of (2.37), we have

$$\begin{aligned}& f(\mathbf{x}_{k+1}^s) - \Phi_{k+1}^s(\mathbf{v}_{k+1}^s) \\ \leq\ & f(\mathbf{x}_k^s) + \eta_k^s\langle \nabla f(\mathbf{x}_k^s), \mathbf{v}_{k+1}^s - \mathbf{x}_k^s\rangle + \frac{(\eta_k^s)^2 L\|\mathbf{v}_{k+1}^s - \mathbf{x}_k^s\|^2}{2} - \Phi_{k+1}^s(\mathbf{v}_{k+1}^s) \\ \overset{(a)}{\leq}\ & (1-\delta_k^s)\big[f(\mathbf{x}_k^s) - \Phi_k^s(\mathbf{v}_k^s)\big] + \frac{(\delta_k^s)^2 L\|\mathbf{v}_{k+1}^s - \mathbf{x}_k^s\|^2}{2} \\ \overset{(b)}{\leq}\ & (1-\delta_k^s)\big[f(\mathbf{x}_k^s) - \Phi_k^s(\mathbf{v}_k^s)\big] + \frac{(\delta_k^s)^2 LD^2}{2}\end{aligned} \tag{2.39}$$

where (a) uses $\eta_k^s = \delta_k^s$ and (2.38); and (b) relies on Assumption 3. For convenience, let us define $\Delta^s(i,j) := \prod_{\tau=i}^{j}(1-\delta_\tau^s)$. Then unrolling (2.39), we get

$$\begin{aligned}& f(\mathbf{x}_{k+1}^s) - \Phi_{k+1}^s(\mathbf{v}_{k+1}^s) \\ \leq\ & \Delta^s(0,k)\big[f(\mathbf{x}_0^s) - \Phi_0^s(\mathbf{v}_0^s)\big] + \sum_{\tau=0}^{k}\frac{LD^2(\delta_\tau^s)^2}{2}\Delta^s(\tau+1,k) \\ \leq\ & \frac{C^s(C^s+1)}{(k+1+C^s)(k+2+C^s)}\big[f(\mathbf{x}_0^s) - \Phi_0^s(\mathbf{v}_0^s)\big] + \frac{2(k+1)LD^2}{(k+1+C^s)(k+2+C^s)}.\end{aligned}$$

When $s=0$, plugging $C^0 = 0$, we have

$$f(\mathbf{x}_{k+1}^0) - \Phi_{k+1}(\mathbf{v}_{k+1}^0) \leq \frac{2LD^2}{k+2}. \tag{2.40}$$

Hence (2.11a) in Theorem 5 is proved. Next consider $s \geq 1$. Using the observation that $f(\mathbf{x}_0^s) - \Phi_0^s(\mathbf{v}_0^s) = \bar{\mathcal{G}}_{K_{s-1}}^{s-1} < \mathcal{G}_{K_{s-1}}^{s-1}$, we then have

$$\begin{aligned}& \mathcal{G}_{k+1}^s = f(\mathbf{x}_{k+1}^s) - \Phi_{k+1}^s(\mathbf{v}_{k+1}^s) \\ <\ & \frac{C^s(C^s+1)}{(k+1+C^s)(k+2+C^s)}\mathcal{G}_{K_{s-1}}^{s-1} + \frac{2(k+1)LD^2}{(k+1+C^s)(k+2+C^s)} \\ \overset{(c)}{=}\ & \frac{2LD^2(C^s+1)}{(k+1+C^s)(k+2+C^s)} + \frac{2(k+1)LD^2}{(k+1+C^s)(k+2+C^s)} = \frac{2LD^2}{k+1+C^s}.\end{aligned} \tag{2.41}$$

where (c) uses the definition of C^s. Hence (2.11b) in Theorem 5 is proved.

Finally, we only need to show that $C^s \geq 1 + \sum_{j=0}^{s-1} K_j$ by induction. First by definition of $C^1 = 2LD^2/(\mathcal{G}_{K_0}^0)$, with $\mathcal{G}_{K_0}^0 \leq \frac{2LD^2}{K_0+1}$, it is clear that $C^1 \geq 1 + K_0$. Then suppose $C^s \geq 1 + \sum_{j=0}^{s-1} K_j$ hold for some s, we will show that $C^{s+1} \geq 1 + \sum_{j=0}^{s} K_j$.

Using (2.41), we have $C^{s+1} = 2LD^2/(\mathcal{G}_{K_s}^s) \geq C^s + K_s \geq 1 + \sum_{j=0}^{s-1} K_j + K_s$. Hence (2.11b) is proved.

For the smooth step size (2.9) and line search (2.10), the same bound can be obtained by using the same arguments as in Theorems 2 and 3. Hence they are omitted here. □

2.5.12 Proof of Corollary 2

Proof. Using Definition 4 and following the standard derivation of descent lemma [8, Lemma 1.2.3], we can show that

$$\begin{aligned} & f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) \qquad (2.42) \\ & \leq \eta_k \langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \rangle + \frac{\eta_k^2 L(\mathbf{x}_k, \mathbf{x}_{k+1})}{2} \|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2 \\ & \leq \eta_k \langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \rangle + \frac{\eta_k^2 L(\mathbf{x}_k, \mathbf{v}_{k+1})}{2} \|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2. \end{aligned}$$

The reason for $L(\mathbf{x}_k, \mathbf{v}_{k+1}) \geq L(\mathbf{x}_k, \mathbf{x}_{k+1})$ is that $\mathbf{x}_{k+1}$ lives in between $\mathbf{x}_k$ and $\mathbf{v}_{k+1}$. Although $L(\mathbf{x}_k, \mathbf{x}_{k+1})$ can provide a tighter bound, it is not tractable.

Combining (2.42) and (2.13), we have

$$\begin{aligned} & f(\mathbf{x}_{k+1}) - \Phi_{k+1}(\mathbf{v}_{k+1}) \qquad (2.43) \\ & \leq (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + (\eta_k - \delta_k)\big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \big\rangle + \frac{\eta_k^2 L(\mathbf{x}_k, \mathbf{v}_{k+1}) \|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2}. \end{aligned}$$

It can be verified that the specific choice of η_k in (2.7) minimizes the RHS of (2.43) over $[0, 1]$.

Hence we have

$$
\begin{aligned}
& f(\mathbf{x}_{k+1}) - \Phi_{k+1}(\mathbf{v}_{k+1}) && (2.44)\\
& \leq (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + \frac{\eta_k^2 L(\mathbf{x}_k, \mathbf{v}_{k+1})\|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2} + (\eta_k - \delta_k)\big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \big\rangle \\
& \overset{(a)}{\leq} (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + \frac{\alpha_k^2 L(\mathbf{x}_k, \mathbf{v}_{k+1})\|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2} + (\alpha_k - \delta_k)\big\langle \nabla f(\mathbf{x}_k), \mathbf{v}_{k+1} - \mathbf{x}_k \big\rangle \\
& \overset{(b)}{=} (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + \frac{\delta_k^2 L(\mathbf{x}_k, \mathbf{v}_{k+1})\|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2} \\
& \overset{(c)}{=} (1-\delta_k)\big[f(\mathbf{x}_k) - \Phi_k(\mathbf{v}_k)\big] + \frac{\delta_k^2 L\|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2}{2} \\
& \leq \frac{2LD^2}{k+2}
\end{aligned}
$$

where in (a) α_k can be chosen as any number in $[0, 1]$; in (b) we set $\alpha_k = \delta_k$; and (c) uses $L(\mathbf{x}_k, \mathbf{v}_{k+1}) \leq L$. This completes the proof. □

2.5.13 Computing directionally smooth constant

Define a one dimensional function $g(\eta) := f\big(\mathbf{x}_k + \eta(\mathbf{v}_{k+1} - \mathbf{x}_k)\big)$, where $\mathrm{dom}\ \eta = [0, 1]$. Then it is clear that $\nabla g(\eta) = \langle \mathbf{v}_{k+1} - \mathbf{x}_k, \nabla f\big(\mathbf{x}_k + \eta(\mathbf{v}_{k+1} - \mathbf{x}_k)\big)\rangle$. Therefore, it is easy to see that $g(\eta)$ is smooth, i.e.,

$$
\begin{aligned}
& \big|\nabla g(\eta_1) - \nabla g(\eta_2)\big| \\
& = |\langle \mathbf{v}_{k+1} - \mathbf{x}_k, \nabla f\big(\mathbf{x}_k + \eta_1(\mathbf{v}_{k+1} - \mathbf{x}_k)\big) - \nabla f\big(\mathbf{x}_k + \eta_2(\mathbf{v}_{k+1} - \mathbf{x}_k)\big)\rangle| \\
& \leq \|\mathbf{v}_{k+1} - \mathbf{x}_k\| \big\|\nabla f\big(\mathbf{x}_k + \eta_1(\mathbf{v}_{k+1} - \mathbf{x}_k)\big) - \nabla f\big(\mathbf{x}_k + \eta_2(\mathbf{v}_{k+1} - \mathbf{x}_k)\big)\big\|_* \\
& \leq L(\mathbf{x}_k, \mathbf{v}_{k+1})\|\mathbf{v}_{k+1} - \mathbf{x}_k\|^2 |\eta_1 - \eta_2| && (2.45)
\end{aligned}
$$

On the other hand, one can also analytically find L_g by definition; i.e., $\big|\nabla g(\eta_1) - \nabla g(\eta_2)\big| \leq L_g\big|\eta_1 - \eta_2\big|$. Comparing L_g with RHS of (2.45), we can obtain $L(\mathbf{x}_k, \mathbf{v}_{k+1})$. This method can be applied when f is e.g., quadratic loss and logistic loss.

2.5.14 Numerical tests on binary classification

All numerical experiments are performed using Python 3.7 on an Intel i7-4790CPU @3.60 GHz (32 GB RAM) desktop.

Table 2.2: A summary of datasets used in numerical tests

Dataset	d	N (train)	nonzeros
w7a	300	$24,692$	3.89%
realsim	$20,958$	$50,617$	0.24%
mushromm	122	$8,124$	18.75%
ijcnn1	22	$49,990$	40.91%

2.5.15 Numerical tests on matrix completion

The dataset used for the test is *MovieLens100K*, where 1682 movies are rated by 943 users with 6.30% ratings observed. The initialization and data processing are the same as those used in [18].

Chapter 3

Nesterov's momentum for parameter-free FW

3.1 Introduction

Although we have shown that heavy ball momentum is useful for improving primal dual error, HFW exhibits slower convergence when compared to Nesterov's accelerated gradient (NAG) method, also known as accelerated gradient method (AGM)[1] , a projection based algorithm converging at $\mathcal{O}(\frac{1}{k^2})$. As we have seen previously, both FW and HFW satisfy $f(\mathbf{x}_k) - f(\mathbf{x}^*) = \mathcal{O}(\frac{1}{k})$. Accelerating the convergence rate for FW or HFW is in general not possible supported by the lower bound in [3, 2]. However, improved FW type algorithms are possible in speedup rates for certain subclasses of problems. In contrary to AGM, this accelerated version of FW does not rely on the smooth parameter L.

3.1.1 Related works

Recall that there are three common approaches to select step sizes for FW and its variants: i) line search [2]; ii) minimizing a one-dimensional quadratic function over $[0, 1]$ for smooth step sizes [45, 55]; and iii) parameter-free step sizes; that is, $\mathcal{O}(\frac{1}{k})$ [2]. Most of the fast converging FW iterations rely on choices i) or ii), which require either the smoothness parameter or the function value of f. Step size i) is 'clumsy' when it is costly to access function values, e.g., in

[1] We will use NAG and AGM interchangeably.

the big data regime. Concerns with choice ii) arise with how well the smoothness parameter is estimated. In addition, it is challenging to select the smoothness inducing norm, and each norm can result in a considerably different smoothness parameter [64]. The need thus arises for FW variants relying on parameter-free step sizes, especially those enabling faster convergence. To this end, we first briefly recap existing results on faster rates.

Line search. Jointly leveraging line search and 'away steps,' FW-type algorithms converge linearly for strongly convex problems when $\mathcal{X}$ is a polytope [43, 42]; see also [65, 66], and [67] where the memory efficiency of away steps is also improved.

Smooth step sizes. If $\mathcal{X}$ is strongly convex, and the optimal solution is at the boundary of $\mathcal{X}$, it is known that FW converges linearly [45]. For uniformly (and thus strongly) convex sets, faster rates are attained when the optimal solution is at the boundary of $\mathcal{X}$ [68]. When both f and $\mathcal{X}$ are strongly convex, FW with the smooth step size converges at a rate of $\mathcal{O}(\frac{1}{k^2})$, regardless of where the optimal solution resides [55]. A variant of smooth step size along with modifications on FW jointly enable faster rates on a strongly convex f and Gauge set $\mathcal{X}$ [69], at the expense of requiring extra parameters besides the smoothness constant.

Parameter-free step sizes. Without any parameter involved here, there is no concern on the quality of parameter estimation, which saves time and effort because there is no need for tuning step sizes. 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 smooth step sizes. Faster rates for parameter-free FW are rather limited in number. In a recent work [70], the behavior of FW when k is large and $\mathcal{X}$ is a polytope is investigated under the strong assumptions on $f(\mathbf{x})$ being twice differentiable and locally strongly convex around $\mathbf{x}^*$. Hence, the analysis does not hold for e.g., the Huber loss, which is widely used in robust regression but is only once-differentiable. The faster rates, along with the assumptions on f and $\mathcal{X}$, are summarized in Table 3.1 for comparison. To establish faster rates, our solution connects the FW subproblem with Nesterov's momentum, which is recapped next.

Nesterov's momentum. After the $\mathcal{O}(\frac{1}{k^2})$ convergence rate was established in [48, 8], the efficiency of Nesterov momentum is proven almost universal; see e.g., the accelerated proximal gradient [71, 72], projected AGM [73, 72] for problems with constraints; accelerated mirror descent [73, 74, 72], and accelerated variance reduction for problems with finite-sum structures [75, 76]. Parallel to these works, AGM has been also investigated from an ordinary differential equation (ODE) perspective [77, 74, 78, 79]. However, the efficiency of Nesterov momentum on FW type algorithms is shaded given the lower bound on the number of subproblems [3, 2]. One idea to introduce momentum into FW is to adopt CGS [49], where the projection subproblem in the original AGM is substituted by gradient sliding which solves a sequence of FW subproblems. The faster rate $\mathcal{O}(\frac{1}{k^2})$ is obtained with the price of: i) the requirement of at most $\mathcal{O}(k)$ FW subproblems in the kth iteration; and ii) an inefficient implementation (e.g., the AGM subproblem has to be solved to certain accuracy, and it relies on other parameters that are not necessary in FW, such as the diameter of $\mathcal{X}$).

Although parameter-free FW is undoubtedly attractive in several applications, there are two main challenges in establishing faster rates for such step sizes: i) even AGM and most of its variants are not parameter-free since they involve a smoothness parameter; and ii) parameter-free FW in general cannot ensure per step descent, which is essential for faster rates. To overcome these challenges, we first unveil the links between the notion of momentum and the FW subproblem. Then, we leverage these connections to provide provable constraint-dependent faster rates.

Table 3.1: A comparison of FW variants with faster rates, where 'ls', 'smooth', and 'pf' are short for line search, smooth step size, and parameter-free step sizes, respectively.

work	**assumptions on f**	**assumptions on $\mathcal{X}$**	**step sizes**	**convergence**
[42, 67]	smooth and strongly convex	polytopes	ls	linear
[45]	smooth and convex	active strongly convex sets, e.g., active ℓ_p norm balls with $p \in (1, 2]$	smooth	linear
[55]	smooth and strongly convex	strongly convex sets	smooth	$\mathcal{O}(\frac{1}{k^2})$
[70]	smooth, convex, twice differentiable, and locally strongly convex around $\mathbf{x}^*$	polytopes	pf	$\mathcal{O}(\frac{1}{k^2})$
This work	smooth and convex	active ℓ_p norm balls with $p \in [1, +\infty)$	pf	$\tilde{\mathcal{O}}(\frac{1}{k^2})$

3.1.2 Our contributions

In succinct form, our contributions are as follows.

- We observe that the momentum update in AGM plays a similar role as the subproblem in FW, intuitively and analytically. Hence, the FW subproblem can be leveraged to play the role of Nesterov's momentum, thus enabling faster rates on a useful family of problems.
- We prove that a momentum-guided FW, termed accelerated Frank Wolfe (AFW), achieves a faster rate $\tilde{\mathcal{O}}(\frac{1}{k^2})$ on active ℓ_p norm ball constraints without knowledge of the smoothness parameter or the function value. We also establish that AFW converges no slower than FW on general problems.
- We corroborate the numerical efficiency of AFW on two benchmark tasks. We validate faster AFW rates on binary classification problems with different constraint sets. We further demonstrate that for matrix completion, AFW finds low-rank solutions with small optimality error more rapidly than FW.

3.2 Connecting Nesterov's momentum with FW

Algorithm 4 AGM [8]

```
Initialize: $\mathbf{x}_0$
for $k = 0, 1, \dots, K-1$ do
    $\mathbf{y}_k = \delta_k \mathbf{v}_k + (1 - \delta_k)\mathbf{x}_k$
    $\mathbf{x}_{k+1} = \mathbf{y}_k - \frac{1}{L}\nabla f(\mathbf{y}_k)$
    $\mathbf{v}_{k+1} = \mathbf{v}_k - \frac{\delta_k}{\mu_{k+1}}\nabla f(\mathbf{y}_k)$
end for
Return: $\mathbf{x}_K$
```

To bring intuition on how momentum can be helpful for FW type algorithms, we first recap AGM for unconstrained convex problems, i.e., $\mathcal{X} = \mathbb{R}^d$. For simplicity, we also assume $\|\cdot\|$ stands for ℓ_2 norm in this section. Note that the reason for discussing the unconstrained problem here is only for the simplicity of exposition, and one can extend the arguments to constrained cases straightforwardly. AGM [48, 8, 73] is summarized in Alg. 4. We start this section by characterizing the behavior of $\{\mathbf{x}_k\}$, $\{\mathbf{y}_k\}$ and $\{\mathbf{v}_k\}$ in the next theorem.

Theorem 8. *Under Assumptions 1 and 2, with $\delta_k = \frac{2}{k+3}$, $\mu_0 = 2L$, and $\mu_{k+1} = (1-\delta_k)\mu_k$, AGM in Alg. 4 guarantees that*

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) = \mathcal{O}\Big(\frac{f(\mathbf{x}_0) - f(\mathbf{x}^*) + L\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{k^2}\Big), \ \forall k.$$
$$\|\nabla f(\mathbf{y}_k)\|^2 \leq \mathcal{O}\bigg(\frac{L\big(f(\mathbf{x}_0) - f(\mathbf{x}^*) + L\|\mathbf{x}_0 - \mathbf{x}^*\|^2\big)}{(k+2)^2}\bigg), \ \forall k.$$

In addition, it holds for any k that $\|\mathbf{v}_k - \mathbf{x}^*\|^2 \leq \frac{1}{L}\big(f(\mathbf{x}_0) - f(\mathbf{x}^*) + L\|\mathbf{x}_0 - \mathbf{x}^*\|^2\big)$.

Theorem 8 shows that $\|\nabla f(\mathbf{y}_k)\|^2 = \mathcal{O}(\frac{1}{k^2})$, which implies that $\mathbf{y}_k$ also converges to a minimizer as $k \to \infty$. Through the increasing step size $\frac{\delta_k}{\mu_{k+1}} = \mathcal{O}(\frac{k}{L})$, the update of $\mathbf{v}_k$ stays in the ball centered at $\mathbf{x}^*$ with radius depending on both $\mathbf{x}^*$ and $\mathbf{x}_0$.

One observation of AGM is that by substituting Line 6 in Alg. 4 with $\mathbf{v}_{k+1} = \mathbf{x}_{k+1}$, the modified algorithm boils down to GD. Hence, it is clear that the key behind AGM's acceleration is $\mathbf{v}_k$ and the way it is updated. We contend that the $\mathbf{v}_{k+1}$ is obtained by minimizing an approximated lower bound of $f(\mathbf{x})$ formed as the summation of a supporting hyperplane at $\mathbf{y}_k$ and a regularizer. To see this, one can rewrite Line 6 of AGM as

$$\mathbf{v}_{k+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^d} \underbrace{f(\mathbf{y}_k) + \langle \nabla f(\mathbf{y}_k), \mathbf{x} - \mathbf{y}_k\rangle}_{\text{supporting hyperplane}} + \underbrace{\frac{\mu_{k+1}}{2\delta_k}\|\mathbf{x} - \mathbf{v}_k\|^2}_{\text{regularizer}} \tag{3.1}$$

where the linear part is the supporting hyperplane, and $\frac{\mu_{k+1}}{\delta_k} = \mathcal{O}(\frac{L}{k})$. As k increases, the impact of the regularizer $\frac{\mu_{k+1}}{2\delta_k}\|\mathbf{x} - \mathbf{v}_k\|^2$ in (3.1) will become limited. Thus the RHS can be viewed as an approximated lower bound of $f(\mathbf{x})$. Regarding the reasons to put a regularizer after the supporting hyperplane, it first guarantees the minimizer *exists* since directly minimize the supporting hyperplane over $\mathbb{R}^d$ yields no solution. In addition, $\mathbf{v}_{k+1}$ is ensured to be *unique* because the RHS of (3.1) is strongly convex thanks to the regularizer. Since $\mathbf{v}_{k+1}$ minimizes an approximated lower bound of $f(\mathbf{x})$, it can be used to estimate $f(\mathbf{x}^*)$. We explain in Theorem 11 in Appendix 3.5.2 that $f(\mathbf{y}_k) + \langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{y}_k\rangle$ approximates $f(\mathbf{x}^*)$. Consequently, one can obtain an estimated suboptimality gap using $f(\mathbf{x}_{k+1}) - f(\mathbf{y}_k) - \langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{y}_k\rangle$.

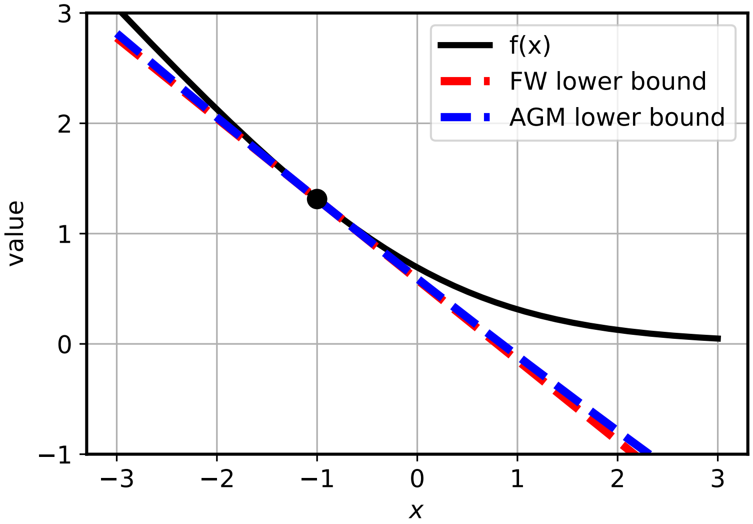


Figure 3.1: Similarity between the RHS of (1.8) and (3.1).

Momentum $\mathbf{v}_k$ update as an FW step. It is observed that $\mathbf{v}_{k+1}$ in both FW and AGM (cf. (1.8) and (3.1)) are obtained by minimizing an (approximated) lower bound of $f(\mathbf{x})$, where the only difference lies on whether a regularizer with decreasing weights is utilized. The similarity between the RHS of (1.8) and (3.1) will be amplified when k is large; see Fig. 3.1 for a graphical illustration on how (3.1) approaches to an affine function. In other words, the momentum update in (3.1) becomes similar to an FW step for a large k. In addition, there are also several other connections.

Connection 1. The $\mathbf{v}_{k+1}$ update via (3.1) is equivalent to

$$\mathbf{v}_{k+1} = \arg\min_{\mathbf{v}\in\mathcal{V}_k} \langle \nabla f(\mathbf{y}_k), \mathbf{v} - \mathbf{y}_k \rangle \tag{3.2}$$

for $\mathcal{V}_k := \{\mathbf{v} | \|\mathbf{v} - \mathbf{v}_k\|^2 \leq r_k\}$ with r_k denoting the time-varying radius of the norm ball. Clearly, r_k depends on $\frac{\mu_{k+1}}{2\delta_k}$, and it is upper bounded by $\frac{2}{L}\big(f(\mathbf{x}_0) - f(\mathbf{x}^*) + L\|\mathbf{x}_0 - \mathbf{x}^*\|^2\big)$ according to Theorem 8. By rewriting (3.1) in its constrained form (3.2), it can be readily recognized that for unconstrained problems *Nesterov momentum can be obtained via FW steps* with *time-varying* constraint sets.

Connection 2. Recall that in AGM, $\mathbf{v}_{k+1}$ obtained via (3.1) is used to construct an approximation of $f(\mathbf{x}^*)$, which is $f(\mathbf{y}_k) + \langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{y}_k \rangle$. When a compact $\mathcal{X}$ is present, directly minimizing the supporting hyperplane $f(\mathbf{y}_k) + \langle \nabla f(\mathbf{y}_k), \mathbf{x} - \mathbf{y}_k \rangle$ over $\mathcal{X}$ also yields an estimate of $f(\mathbf{x}^*)$. Note that the latter is exactly an FW step. In addition, the FW step in Alg. 1 also results in a suboptimality gap (known as FW gap; see e.g., [2]), which is in line with the role of $\mathbf{v}_k$ in AGM. In a nutshell, both FW step and momentum update in AGM result in an estimated suboptimality gap.

Connection 3. Connections between momentum and FW go beyond convexity. We discuss in Appendix 3.5.3 that AGM for strongly convex problems updates its momentum using exactly the same idea of FW, that is, both obtain a minimizer of a lower bound of $f(\mathbf{x})$, and then perform an update through a convex combination.

These links and similarities between momentum and FW naturally lead us to explore their connections, and see how momentum influences FW.

3.3 Momentum-guided FW

Algorithm 5 AFW

```
Initialize: $\mathbf{x}_0 \in \mathcal{X}$, $\boldsymbol{\theta}_0 = \mathbf{0}$
for $k = 0, 1, \ldots, K-1$ do
    $\mathbf{y}_k = (1-\delta_k)\mathbf{x}_k + \delta_k \mathbf{v}_k$
    $\boldsymbol{\theta}_{k+1} = (1-\delta_k)\boldsymbol{\theta}_k + \delta_k \nabla f(\mathbf{y}_k)$
    $\mathbf{v}_{k+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \langle \boldsymbol{\theta}_{k+1}, \mathbf{x} \rangle$
    $\mathbf{x}_{k+1} = (1-\delta_k)\mathbf{x}_k + \delta_k \mathbf{v}_{k+1}$
end for
Return: $\mathbf{x}_K$
```

In this section we show that the momentum is beneficial for FW by proving that it is effective at least on certain constraint sets. Specifically, we will focus on the accelerated Frank Wolfe (AFW) summarized in Alg. 5, and analyze its convergence rate. Since we will see later that $\delta_k = \frac{2}{k+3} \in (0,1)$, $\forall k$, for which $\mathbf{y}_k$, $\mathbf{v}_k$ and $\mathbf{x}_k$ lie in $\mathcal{X}$ for all k, AFW is projection free. Albeit rarely, it is safe to choose $\mathbf{v}_{k+1} = \mathbf{v}_k$, and proceed when $\boldsymbol{\theta}_{k+1} = \mathbf{0}$. Note that the $\mathbf{x}_{k+1}$ update in AFW is slightly different with that of AGM. This is because AGM guarantees $f(\mathbf{x}_{k+1}) \leq f(\mathbf{y}_k)$, $\forall k$, taking advantage of the known L. However, the same guarantee is difficult to be replicated in a parameter-free algorithm.

The key to AFW is the $\mathbf{v}_{k+1}$ update, which plays the role of momentum. To see this, if one unrolls $\boldsymbol{\theta}_{k+1}$ (cf. (3.17) in Appendix) and plugs it into Line 5 of Alg. 5, $\mathbf{v}_{k+1}$ can be equivalently rewritten as

$$\mathbf{v}_{k+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \sum_{\tau=0}^{k} w_k^\tau \big[f(\mathbf{y}_\tau) + \langle \nabla f(\mathbf{y}_\tau), \mathbf{x} - \mathbf{y}_\tau \rangle\big] \tag{3.3}$$

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 exact value of the sum depends on the choice of δ_τ). 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 the RHS of (3.3) is a lower bound for $f(\mathbf{x})$ constructed through a weighted average of supporting hyperplanes at $\{\mathbf{y}_\tau\}$. In other words, $\mathbf{v}_{k+1}$ is a minimizer of a lower bound of $f(\mathbf{x})$, hence it is in line with the role of momentum. However, the momentum in AFW differs from AGM in two aspects. First, instead of relying on $\nabla f(\mathbf{y}_k)$, the update of $\mathbf{v}_{k+1}$ utilizes coefficient $\boldsymbol{\theta}_{k+1}$, which is (roughly) a weighted average of past gradients $\{\nabla f(\mathbf{y}_\tau)\}_{\tau=1}^{k}$ with more weight placed on recent ones. The second difference on the $\mathbf{v}_{k+1}$ update with AGM is whether a regularizer is used. As a consequence of the non-regularized lower bound (3.3), its minimizer is *not* guaranteed to be unique. A simple example is to consider the ith entry $[\boldsymbol{\theta}_{k+1}]_i = 0$. The ith entry $[\mathbf{v}_{k+1}]_i$ can then be chosen arbitrarily as long as $\mathbf{v}_{k+1} \in \mathcal{X}$. This subtle difference leads to a significant gap between the performance of AFW and AGM, that is, AFW cannot achieve acceleration on general problems, as will be illustrated shortly. However, we confirm that momentum is still helpful since it is effective on a class of problems.

3.3.1 AFW convergence for general problems

The analysis of AFW relies on a tool known as estimate sequence (ES) introduced by [8]. ES is commonly adopted to analyze projection based algorithms; see e.g., [75, 76, 80, 81, 64], but seldomly used for FW. Formally, ES is defined as follows.

Definition 5. *(ES.) 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* $\lim_{k\rightarrow\infty}\lambda_k = 0$*, and for any* $\mathbf{x}\in\mathbb{R}^d$ *we have*

$$\Phi_k(\mathbf{x}) \leq (1-\lambda_k)f(\mathbf{x}) + \lambda_k \Phi_0(\mathbf{x}).$$

ES is generally not unique and different constructions can be used to design different algorithms. To highlight our analysis technique, recall that quadratic surrogate functions $\{\Phi_k(\mathbf{x})\}$ are used for the derivation of AGM [8] (or see (3.7) in Appendix). Different from AGM, and taking advantage of the compact constraint set, here we consider *linear* surrogate functions for AFW

$$\Phi_0(\mathbf{x}) \equiv f(\mathbf{x}_0) \tag{3.4a}$$

$$\Phi_{k+1}(\mathbf{x}) = (1-\delta_k)\Phi_k(\mathbf{x}) + \delta_k\Big[f(\mathbf{y}_k) + \big\langle \nabla f(\mathbf{y}_k), \mathbf{x}-\mathbf{y}_k \big\rangle\Big], \ \forall\, k \geq 0. \tag{3.4b}$$

Evidenced by the terms in the bracket of (3.4b), i.e., it is a supporting hyperplane of $f(\mathbf{x})$, $\Phi_{k+1}(\mathbf{x})$ is an approximated lower bound of $f(\mathbf{x})$ constructed by weighting the supporting hyperplanes at $\{\mathbf{y}_\tau\}_{\tau=0}^{k}$. Next, we show that (3.4) together with proper $\{\lambda_k\}$ forms an ES for f. Through the ES based proof, it is also revealed that the link between the momentum in AGM and the FW step is also in the technical proof level.

Lemma 3. *With $\lambda_0 = 1$ and $\lambda_k = \lambda_{k-1}(1-\delta_{k-1})$, the tuple $\big(\{\Phi_k(\mathbf{x})\}_{k=0}^{\infty}, \{\lambda_k\}_{k=0}^{\infty}\big)$ in* (3.4) *is an ES of $f(\mathbf{x})$.*

Using properties of the functions in (3.4) (cf. Lemma 6 in Appendix 3.5.5), the following lemma holds for AFW.

Lemma 4. *With $\Phi_k^* := \min_{\mathbf{x}\in\mathcal{X}} \Phi_k(\mathbf{x})$, AFW is guaranteed to satisfy $f(\mathbf{x}_{k+1}) \leq \Phi_{k+1}^* + \xi_{k+1}$, $\forall\, k$, where $\xi_{k+1} = (1-\delta_k)\xi_k + \frac{L\delta_k^2}{2}\|\mathbf{v}_{k+1} - \mathbf{v}_k\|^2$ and $\xi_0 = 0$.*

Leveraging Lemma 4, the convergence rate of AFW for general problems can be established.

Theorem 9. *When Assumptions 1, 2 and 3 are satisfied, upon choosing $\delta_k = \frac{2}{k+3}$ and $\boldsymbol{\theta}_0 = \mathbf{0}$, AFW guarantees*

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{2\big(f(\mathbf{x}_0) - f(\mathbf{x}^*)\big)}{(k+1)(k+2)} + \frac{2LD^2}{k+2}, \ \forall\, k.$$

Theorem 9 asserts that the convergence rate of AFW is $\mathcal{O}(\frac{LD^2}{k})$, coinciding with that of FW [2]. Notwithstanding, AFW is tight in terms of the number of FW steps required. To see this, note that the convergence rate in Theorem 9 translates to requiring $\mathcal{O}(\frac{LD^2}{\epsilon})$ FW steps to guarantee $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \epsilon$. This matches the lower bound [2, 51]. Similar to other FW variants, acceleration for AFW cannot be claimed for general problems. AFW however, is attractive numerically because it can alleviate the zig-zag behavior[2] of FW [82], as we will see in Section 3.4.

[2] The change between $f(\mathbf{x}_{k+1})$ and $f(\mathbf{x}_k)$ is large with high frequency, so zig-zag emerges when plotting $f(\mathbf{x}_k) - f(\mathbf{x}^*)$ versus k.

Why acceleration cannot be achieved in general? Recall from Lemma 4, that critical to acceleration is ensuring a small ξ_k, which in turn requires $\mathbf{v}_{k+1}$ and $\mathbf{v}_k$ to stay sufficiently close. This is difficult in general because the non-uniqueness of $\mathbf{v}_k$ prevents one from ensuring a small upper bound of $\|\mathbf{v}_k - \mathbf{v}_{k+1}\|^2 \; \forall \; \mathbf{v}_k, \forall \; \mathbf{v}_{k+1}$. The ineffectiveness of momentum in AFW in turn signifies the importance of the added regularizer in AGM momentum update (3.1).

3.3.2 AFW acceleration for a class of problems

In this subsection, we provide constraint-dependent accelerated rates of AFW when $\mathcal{X}$ is a ball induced by some norm. Even for projection based algorithms, most accelerated rates are obtained with L-dependent step sizes [83]. Thus, faster rates for parameter-free algorithms are challenging to establish. An extra assumption is needed in this subsection.

Assumption 4. *The constraint is active; that is,* $\|\nabla f(\mathbf{x}^*)\|_2^2 \geq G > 0$.

To analyze convergence of FW iterations, it is reasonable to rely on the position of the optimal solution, which justifies why this assumption is also adopted in [45, 84, 68]. For a number of signal processing and machine learning tasks, Assumption 4 is rather mild. Relying on Lagrangian duality, it can be seen that problem (1.1) 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 4 simply requires $\gamma > 0$ in the equivalent regularized formulation, that is, the norm ball constraint plays the role of a regularizer. Given the prevalence of regularized formulations, it is worth investigating their equivalent constrained form (1.1) under Assumption 4. Next, we will use the ℓ_2 norm ball constraints to illustrate the intuition behind the acceleration.

ℓ_2 **norm ball constraint.** Consider $\mathcal{X} := \{\mathbf{x} | \|\mathbf{x}\|_2 \leq \frac{D}{2}\}$. In this case, $\mathbf{v}_{k+1}$ admits a closed-form solution

$$\mathbf{v}_{k+1} = \arg\min_{\mathbf{x} \in \mathcal{X}} \langle \boldsymbol{\theta}_{k+1}, \mathbf{x} \rangle = -\frac{D}{2\|\boldsymbol{\theta}_{k+1}\|_2} \boldsymbol{\theta}_{k+1}. \tag{3.5}$$

The uniqueness of $\mathbf{v}_{k+1}$ is ensured by its closed-form solution, wiping out the obstacle for a faster rate. In addition, through (3.5) it becomes possible to guarantee that $\mathbf{v}_{k+1}$ and $\mathbf{v}_k$ are close whenever $\boldsymbol{\theta}_k$ is close to $\boldsymbol{\theta}_{k+1}$.

Theorem 10. *If Assumptions 1, 2, 3 and 4 are satisfied, and $\mathcal{X}$ is an ℓ_2 norm ball, choosing $\delta_k = \frac{2}{k+3}$ and $\boldsymbol{\theta}_0 = \mathbf{0}$, AFW guarantees acceleration with convergence rate*

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) = \mathcal{O}\bigg(\min \Big\{ \frac{LD^2T + C\ln k}{k^2}, \frac{LD^2}{k} \Big\} \bigg)$$

where C and T are constants depending on L, D and G.

Theorem 10 demonstrates that momentum improves the convergence of FW by providing a faster rate. Roughly speaking, when the iteration number $k \geq T$, the rate of AFW dominates that of FW. We note that this matches our intuition, that is, the momentum in AGM (3.1) only behaves like an affine function when k is large (so that the weight on the regularizer is small). In addition, the rate in Theorem 10 can be written compactly as $\tilde{\mathcal{O}}\big(\frac{TLD^2}{k^2}\big)$, $\forall k$, hence it achieves acceleration with a worse dependence on D compared to vanilla FW. Note that the choice for δ_k and $\boldsymbol{\theta}_0$ remains the same as those used in general problems, leading to an identical implementation to non-accelerated cases. Compared with CGS, AFW sacrifices the D dependence in the convergence rate to trade for i) the nonnecessity of the knowledge of L and D, and ii) ensuring only one FW subproblem per iteration (whereas at most $\mathcal{O}(k)$ subproblems are needed in CGS).

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. Taking $\mathbf{v}_{k+1}$ as an example, we have

$$\begin{aligned} \mathbf{v}_{k+1} = R \cdot [0, \ldots, 0, -\text{sgn}[\boldsymbol{\theta}_{k+1}]_i, 0, \ldots, 0]^\top \\ \text{with } \ i = \arg\max_j |[\boldsymbol{\theta}_{k+1}]_j|. \end{aligned} \tag{3.6}$$

We show in the Appendix (Theorem 15) that when Assumption 4 holds and the set $\arg\max_j \big|[\nabla f(\mathbf{x}^*)]_j\big|$ has cardinality 1, a faster rate $\mathcal{O}(\frac{T_1LD^2}{k^2})$ can be obtained. The additional assumption here is known as *strict complementarity*, and has been adopted also in, e.g.,[85, 86] for analysis.

ℓ_p **norm ball constraint.** Consider an active ℓ_p norm ball constraint $\mathcal{X} := \{\mathbf{x} | \|\mathbf{x}\|_p \leq R\}$, where $p \in (1, +\infty)$ and $p \neq 2$. The i-th entry of $\mathbf{v}_{k+1}$ is found in closed form as

$$[\mathbf{v}_{k+1}]_i = -[\boldsymbol{\theta}_{k+1}]_i \frac{\big|[\boldsymbol{\theta}_{k+1}]_i\big|^{q-2}}{\|\boldsymbol{\theta}_{k+1}\|_q^{q-1}} \cdot R$$

where $1/p + 1/q = 1$. We discuss in Appendix 3.5.11 that faster rates are possible under mild conditions. Though not covering all cases, it still showcases that the momentum is partially helpful for parameter-free FW algorithms.

Beyond ℓ_p norm balls. In general, when a specific structure of $\mathbf{x}^*$ (e.g., sparsity) is promoted by $\mathcal{X}$ (so that $\mathbf{x}^*$ is likely to live on the boundary), and one can ensure the uniqueness of $\mathbf{v}_k$ through either a closed-form solution or a specific implementation, acceleration can be effected. A direct extension of the results in this subsection to matrix space is when the constraint is a Schatten ℓ_p norm ball. This is because $\|\mathbf{X}\|_p := \|\sigma_1(\mathbf{X}), \sigma_2(\mathbf{X}), \ldots, \sigma_r(\mathbf{X})\|_p$, where $\sigma_i(\mathbf{X})$ denotes the ith singular value of $\mathbf{X}$. Our numerical results confirm the acceleration in Section 3.4.2.

Table 3.2: A summary of datasets used in numerical tests

Dataset	d	n (train)	nonzeros
a9a	123	$32,561$	11.28%
covtype	54	$406,709$	22.12%
mushroom	122	$8,124$	18.75%
mnist (digit 4)	784	$60,000$	12.4%

3.4 Numerical tests

We validate our theoretical findings as well as the efficiency of AFW on two benchmarked machine learning problems, binary classification and matrix completion in this section. All numerical experiments are performed using Python 3.7 on a desktop equipped with Intel i7-4790 CPU @3.60 GHz (32 GB RAM).

3.4.1 Binary classification

We first test the performance of Alg. 5 on binary classification using logistic regression in Section 1.2.2. Datasets from LIBSVM[3] are used in the numerical tests presented. Details regarding the datasets are summarized in Table 3.2. The constraint sets considered include ℓ_1 and ℓ_2 norm balls. As benchmarks, the chosen algorithms are: projected GD with the standard step size $\frac{1}{L}$; parameter-free FW with step size $\frac{2}{k+2}$ [2]; and projected AGM with parameters according to [73]. The step size of AFW is $\delta_k = \frac{2}{k+3}$ according to Theorems 9 and 10. Note that both GD and AGM are not parameter-free.

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

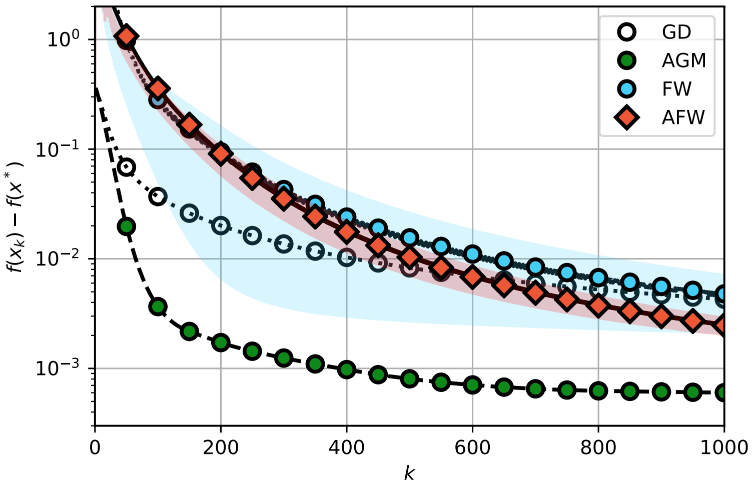
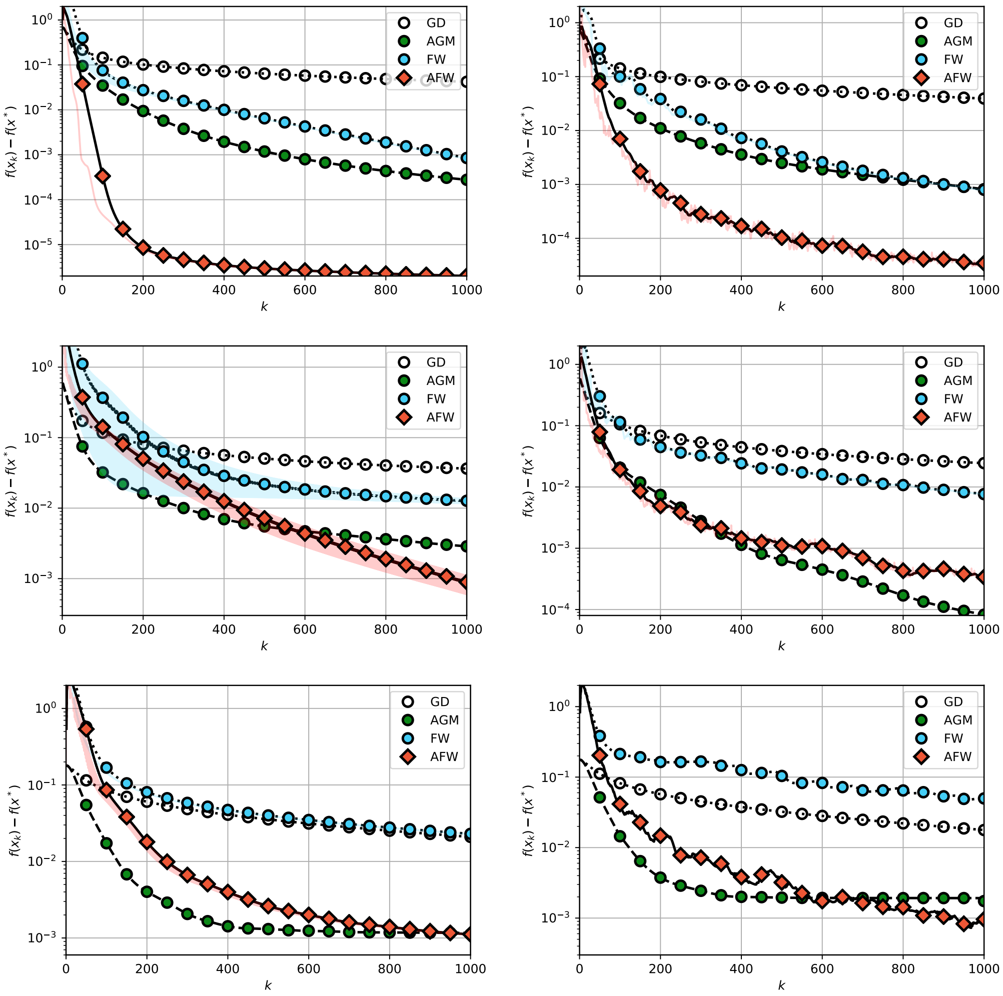


Figure 3.2: Performance of AFW when the optimal solution is at interior.

We first let $\mathcal{X}$ be an ℓ_2 norm ball with a large enough radius so that $\mathbf{x}^*$ does not lie on the boundary. This case maps to our result in Theorem 9, where the convergence rate of AFW is $\mathcal{O}(\frac{1}{k})$. The performance of AFW is shown in Fig. 3.2. On dataset *a9a*, AFW slightly outperforms GD and FW, but is slower than AGM. Evidently, AFW is much more *stable* than FW, as one can see from the shaded areas that illustrate the range of zig-zag.

Next, we consider active ℓ_2 norm ball constraints. In this case, our result in Theorem 10 applies and AFW achieves an $\tilde{\mathcal{O}}(\frac{1}{k^2})$ convergence rate. The performance of AFW is listed in the first row of Fig. 3.3. In all tested datasets, AFW significantly improves over FW, while on datasets other than *covtype*, AFW also outperforms AGM, especially on *mushroom*.

When the constraint set is an ℓ_1 norm ball, the performance of AFW is depicted in the second row of Fig. 3.3. It can be seen that on datasets such as *covtype* and *mnist*, AFW exhibits performance similar to AGM, which is significantly faster than FW. While on dataset *mushroom*, AFW converges even faster than AGM. Note that comparing AFW with AGM is not fair since each FW step requires d operations at most, while projection onto an ℓ_1 norm ball in [87] takes cd operations for some $c > 1$. This means that for the same running time, AFW will run more iterations than AGM. We stick to this unfair comparison to highlight how the optimality error of AFW and AGM evolves with k.



(a) ℓ_2 norm ball

(b) ℓ_1 norm ball

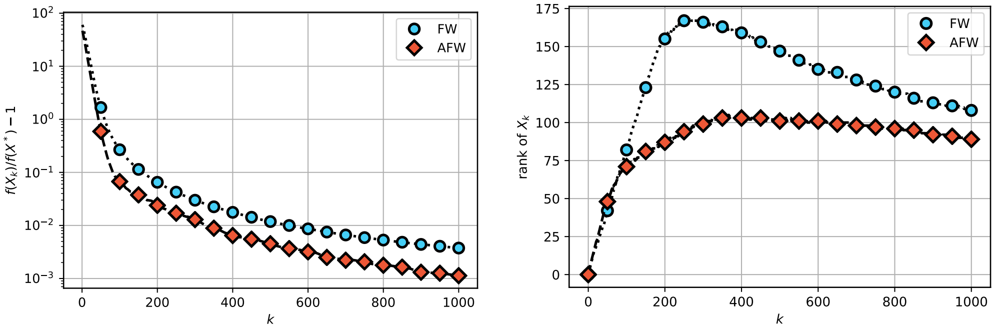
Figure 3.3: Performance of AFW on datasets: *mushroom* (first row), *mnist* (second row), and *covtype* (third row).

3.4.2 Matrix completion

We test AFW and FW on a widely used dataset, *MovieLens100K*[4] , where 1682 movies are rated by 943 users with 6.30% percent ratings observed. And the initialization and data processing are the same as those used in [18]. The numerical performance can be found in Fig. 3.4. In subfigures (a) and (b), we plot the optimality error and rank versus k choosing $R = 3$. It is

[4] Online available at `https://grouplens.org/datasets/movielens/100k/`

observed that AFW exhibits improvement in terms of both optimality error and rank of the solution. In particular, AFW roughly achieves 1.4x performance improvement compared with FW in terms of optimality error, and finds solutions with much lower rank.



(a) optimality

(b) rank

Figure 3.4: Performance of AFW for matrix completion problems.

3.5 Appendix

3.5.1 Proof of Theorem 8

The convergence on $\mathbf{x}_k$ is given in [83], and hence we do not repeat here. Next we show the behavior of $\mathbf{y}_k$ and $\mathbf{v}_k$.

We use the same surrogate functions with those in [83], i.e.,

$$\Phi_0(\mathbf{x}) = \Phi_0^* + \frac{\mu_0}{2}\|\mathbf{x} - \mathbf{x}_0\|^2 \tag{3.7a}$$

$$\Phi_{k+1}(\mathbf{x}) = (1 - \delta_k)\Phi_k(\mathbf{x}) + \delta_k\Big[f(\mathbf{y}_k) + \big\langle \nabla f(\mathbf{y}_k), \mathbf{x} - \mathbf{y}_k \big\rangle\Big], \ \forall\, k \geq 0. \tag{3.7b}$$

In [83], it is shown that with $\lambda_0 = 1$ and $\lambda_k = \lambda_{k-1}(1-\delta_{k-1})$, the tuple $\big(\{\Phi_k(\mathbf{x})\}_{k=0}^{\infty}, \{\lambda_k\}_{k=0}^{\infty}\big)$ is an ES of $f(\mathbf{x})$. In addition, it is also shown that $\Phi_{k+1}(\mathbf{x})$ can be rewritten as $\Phi_k(\mathbf{x}) = \Phi_k^* + \frac{\mu_k}{2}\|\mathbf{x} - \mathbf{v}_k\|^2$, where $\mu_{k+1} = (1 - \delta_k)\mu_k$, and $f(\mathbf{x}_k) \leq \Phi_k^* = \min_{\mathbf{x}} \Phi_k(\mathbf{x})$. We will use these conclusions directly. Rearranging the terms in $\Phi_k(\mathbf{x}) = \Phi_k^* + \frac{\mu_k}{2}\|\mathbf{x} - \mathbf{v}_k\|^2$, we arrive at

$$\begin{aligned}
\frac{1}{2}\|\mathbf{x} - \mathbf{v}_k\|^2 &= \frac{1}{\mu_k}\Big(\Phi_k(\mathbf{x}) - \Phi_k^*\Big) \\
&= \frac{1}{\mu_k}\Big(\Phi_k(\mathbf{x}) - f(\mathbf{x}) + f(\mathbf{x}) - \Phi_k^*\Big) \\
&\overset{(a)}{\leq} \frac{\lambda_k}{\mu_k}\big[\Phi_0(\mathbf{x}) - f(\mathbf{x})\big] + \frac{1}{\mu_k}\big[f(\mathbf{x}) - f(\mathbf{x}_k)\big] \\
&= \frac{1}{2L}\big[\Phi_0(\mathbf{x}) - f(\mathbf{x})\big] + \frac{1}{\mu_k}\big[f(\mathbf{x}) - f(\mathbf{x}_k)\big]
\end{aligned}$$

where (a) is because $\Phi_k(\mathbf{x}) - f(\mathbf{x}) \leq \lambda_k\big(\Phi_0(\mathbf{x}) - f(\mathbf{x})\big)$ by Definition 5, and $f(\mathbf{x}_k) \leq \Phi_k^*$ shown in [8]. Choosing $\mathbf{x}$ as $\mathbf{x}^*$, we arrive at

$$\begin{aligned}
\frac{1}{2}\|\mathbf{x}^* - \mathbf{v}_k\|^2 &\leq \frac{1}{2L}\big[\Phi_0(\mathbf{x}^*) - f(\mathbf{x}^*)\big] - \frac{1}{\mu_k}\big[f(\mathbf{x}_k) - f(\mathbf{x}^*)\big] \\
&\leq \frac{1}{2L}\big[\Phi_0(\mathbf{x}^*) - f(\mathbf{x}^*)\big], \ \forall k.
\end{aligned}$$

This further implies

$$\|\mathbf{x}^* - \mathbf{v}_k\|^2 \leq \frac{1}{L}\big[\Phi_0(\mathbf{x}^*) - f(\mathbf{x}^*)\big], \ \forall k. \tag{3.8}$$

Hence the behavior of $\mathbf{v}_k$ in Theorem 8 is proved.

To prove the convergence of $\mathbf{y}_k$, the following inequality is true as a result of (3.8)

$$\begin{aligned}
\|\mathbf{v}_{k+1} - \mathbf{v}_k\| &\leq \|\mathbf{v}_{k+1} - \mathbf{x}^*\| + \|\mathbf{x}^* - \mathbf{v}_k\| \\
&\leq 2\sqrt{\frac{1}{L}\big[\Phi_0(\mathbf{x}^*) - f(\mathbf{x}^*)\big]}.
\end{aligned}$$

Next, we link $\nabla f(\mathbf{y}_k)$ and $\mathbf{v}_{k+1} - \mathbf{v}_k$ through the update $\mathbf{v}_{k+1} = \mathbf{v}_k - \frac{\delta_k}{\mu_{k+1}}\nabla f(\mathbf{y}_k)$ to get

$$\begin{aligned}
\|\mathbf{v}_{k+1} - \mathbf{v}_k\|^2 &= \frac{(k+2)^2}{4L^2}\|\nabla f(\mathbf{y}_k)\|^2 \\
&\leq \frac{4}{L}\big[\Phi_0(\mathbf{x}^*) - f(\mathbf{x}^*)\big], \ \forall k.
\end{aligned}$$

Rearranging the terms we can obtain the convergence of $\|\nabla f(\mathbf{y}_k)\|^2$, that is,

$$\|\nabla f(\mathbf{y}_k)\|^2 \leq \frac{16L}{(k+2)^2}\big[\Phi_0(\mathbf{x}^*) - f(\mathbf{x}^*)\big].$$

Plugging $\Phi_0(\mathbf{x}^*) = f(\mathbf{x}_0) + L\|\mathbf{x}_0 - \mathbf{x}^*\|^2$ in completes the proof.

3.5.2 $f(\mathbf{y}_k) + \langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{y}_k \rangle$ approximates $f(\mathbf{x}^*)$

We show next that a weighted version of $f(\mathbf{y}_k) + \langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{y}_k \rangle$ is no larger then $f(\mathbf{x}^*) + \mathcal{O}(\frac{1}{k^2})$ to elaborate that $f(\mathbf{y}_k) + \langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{y}_k \rangle$ is (almost) an under-estimate of $f(\mathbf{x}^*)$.

Theorem 11. *If Assumptions 1 and 2 hold, and we choose $\frac{\mu_{k+1}}{\delta_k} = \frac{2L}{k+2}$; and per iteration k, we let $w_k^{(\tau)} = \frac{2(\tau+2)}{k(k+3)}$ for $\tau = 0, 1, \ldots, k-1$, then i) $\sum_{\tau=0}^{k-1} w_k^{(\tau)} = 1$; and, ii)*

$$\sum_{\tau=0}^{k-1} w_k^{(\tau)} \Big[f(\mathbf{y}_\tau) + \langle \nabla f(\mathbf{y}_\tau), \mathbf{v}_{\tau+1} - \mathbf{y}_\tau \rangle \Big] - f(\mathbf{x}^*) \le \frac{2L\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{k(k+3)}.$$

Proof. It is easy to verify that $\sum_{\tau=0}^{k-1} w_k^{(\tau)} = 1$. Next we have

$$\begin{aligned}
& f(\mathbf{y}_k) + \langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{y}_k \rangle \\
= {} & f(\mathbf{y}_k) + \langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{x}^* \rangle + \langle \nabla f(\mathbf{y}_k), \mathbf{x}^* - \mathbf{y}_k \rangle \\
\overset{(a)}{\le} {} & f(\mathbf{x}^*) + \langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{x}^* \rangle \\
= {} & f(\mathbf{x}^*) + \frac{\mu_{k+1}}{\delta_k} \langle \mathbf{v}_k - \mathbf{v}_{k+1}, \mathbf{v}_{k+1} - \mathbf{x}^* \rangle \\
\overset{(b)}{=} {} & f(\mathbf{x}^*) + \frac{\mu_{k+1}}{2\delta_k} \Big[\|\mathbf{x}^* - \mathbf{v}_k\|^2 - \|\mathbf{x}^* - \mathbf{v}_{k+1}\|^2 - \|\mathbf{v}_{k+1} - \mathbf{v}_k\|^2 \Big] \\
\overset{(c)}{=} {} & f(\mathbf{x}^*) + \frac{L}{k+2} \Big[\|\mathbf{x}^* - \mathbf{v}_k\|^2 - \|\mathbf{x}^* - \mathbf{v}_{k+1}\|^2 - \|\mathbf{v}_{k+1} - \mathbf{v}_k\|^2 \Big] \qquad (3.9)
\end{aligned}$$

where (a) follows from the convexity of f, that is, $\langle \nabla f(\mathbf{y}_k), \mathbf{x}^* - \mathbf{y}_k \rangle \le f(\mathbf{x}^*) - f(\mathbf{y}_k)$; (b) uses $2\langle \mathbf{a}, \mathbf{b} \rangle = \|\mathbf{a} + \mathbf{b}\|^2 - \|\mathbf{a}\|^2 - \|\mathbf{b}\|^2$; and (c) is by plugging the value of $\frac{\mu_{k+1}}{\delta_k}$ in. Now, if we define $d_k := f(\mathbf{y}_k) + \langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{y}_k \rangle - f(\mathbf{x}^*)$, rearranging (3.9), we get

$$\begin{aligned}
(k+2)d_k &\le L\Big[\|\mathbf{x}^* - \mathbf{v}_k\|^2 - \|\mathbf{x}^* - \mathbf{v}_{k+1}\|^2 \Big] - L\|\mathbf{v}_{k+1} - \mathbf{v}_k\|^2 \\
&\le L\Big[\|\mathbf{x}^* - \mathbf{v}_k\|^2 - \|\mathbf{x}^* - \mathbf{v}_{k+1}\|^2 \Big]
\end{aligned}$$

Summing over k (and recalling $\mathbf{v}_0 = \mathbf{x}_0$), we arrive at

$$\sum_{\tau=0}^{k-1} (\tau+2) d_\tau \le L\Big[\|\mathbf{x}^* - \mathbf{v}_0\|^2 - \|\mathbf{x}^* - \mathbf{v}_k\|^2 \Big] \le L\|\mathbf{x}^* - \mathbf{x}_0\|^2.$$

By the definition of $w_k^{(\tau)}$, which is $w_k^{(\tau)} = \frac{2(\tau+2)}{k(k+3)}$, we obtain

$$\sum_{\tau=0}^{k-1} w_k^{(\tau)} d_\tau \le \frac{2L\|\mathbf{x}^* - \mathbf{x}_0\|^2}{k(k+3)} \qquad (3.10)$$

which completes the proof. □

3.5.3 AGM links with FW in strongly convex case

We showcase the connection between the momentum update of AGM in strongly convex case and FW. We first formally define strong convexity, which is used in this subsection only.

Assumption 5. *(Strong convexity.) The function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is μ-strongly convex; that is, $f(\mathbf{y}) - f(\mathbf{x}) \geq \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2}\|\mathbf{y} - \mathbf{x}\|^2$, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.*

Under Assumptions 1 and 5, the condition number of f is $\kappa := \frac{L}{\mu}$. To cope with strongly convex problems, Lines 4 – 6 in AGM (Alg. 4) should be modified to [8]

$$\mathbf{y}_k = \frac{1}{1+\delta}\mathbf{x}_k + \frac{\delta}{1+\delta}\mathbf{v}_k \tag{3.11a}$$

$$\mathbf{x}_{k+1} = \mathbf{y}_k - \frac{1}{L}\nabla f(\mathbf{y}_k) \tag{3.11b}$$

$$\mathbf{v}_{k+1} = (1-\delta)\mathbf{v}_k + \delta\mathbf{y}_k - \frac{\delta}{\mu}\nabla f(\mathbf{y}_k). \tag{3.11c}$$

where $\delta = \frac{1}{\sqrt{\kappa}}$. Here $\mathbf{v}_{k+1}$ in (3.11c) denotes the momentum and thus plays the critical role for acceleration. To see how $\mathbf{v}_{k+1}$ is linked with FW, we will rewrite $\mathbf{v}_{k+1}$ as

$$\begin{aligned} \mathbf{z}_{k+1} &= \arg\min_{\mathbf{x}} f(\mathbf{y}_k) + \langle \nabla f(\mathbf{y}_k), \mathbf{x} - \mathbf{y}_k \rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}_k\|^2 \\ &= \mathbf{y}_k - \frac{1}{\mu}\mathbf{y}_k \end{aligned} \tag{3.12a}$$

$$\mathbf{v}_{k+1} = (1-\delta)\mathbf{v}_k + \delta\mathbf{z}_{k+1} \tag{3.12b}$$

Notice that $\mathbf{z}_{k+1}$ is the minimizer of a lower bound of $f(\mathbf{x})$ (due to strongly convexity). Therefore, the $\mathbf{v}_{k+1}$ update is similar to FW in the sense that it first minimizes a lower bound of $f(\mathbf{x})$, then update through convex combination (cf Alg. 1). This demonstrates that the momentum update in AGM shares the same idea of FW update.

3.5.4 Proof of Lemma 3.

Proof. We show this 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

$$
\begin{aligned}
\Phi_{k+1}(\mathbf{x}) &= (1-\delta_k)\Phi_k(\mathbf{x}) + \delta_k\Big[f(\mathbf{y}_k) + \big\langle \nabla f(\mathbf{y}_k), \mathbf{x}-\mathbf{y}_k\big\rangle\Big] \\
&\overset{(a)}{\leq} (1-\delta_k)\Phi_k(\mathbf{x}) + \delta_k f(\mathbf{x}) \\
&\leq (1-\delta_k)\Big[(1-\lambda_k)f(\mathbf{x}) + \lambda_k\Phi_0(\mathbf{x})\Big] + \delta_k f(\mathbf{x}) \\
&= (1-\lambda_{k+1})f(\mathbf{x}) + \lambda_{k+1}\Phi_0(\mathbf{x})
\end{aligned}
$$

where (a) is because the convexity of f; and the last equation is by definition of λ_{k+1}. Together with the fact that $\lim_{k\to\infty}\lambda_k = 0$, the tuple $\big(\{\Phi_k(\mathbf{x})\}_{k=0}^{\infty}, \{\lambda_k\}_{k=0}^{\infty}\big)$ satisfies the definition of an estimate sequence. □

3.5.5 A few useful lemmas.

Lemma 5. *For* $\{\Phi_k(\mathbf{x})\}$ *in* (3.4)*, if* $f(\mathbf{x}_k) \leq \min_{\mathbf{x}\in\mathcal{X}}\Phi_k(\mathbf{x}) + \xi_k$*, it is true that*

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \lambda_k\big(f(\mathbf{x}_0) - f(\mathbf{x}^*)\big) + \xi_k, \ \forall\, k.$$

Proof. If $f(\mathbf{x}_k) \leq \min_{\mathbf{x}\in\mathcal{X}}\Phi_k(\mathbf{x}) + \xi_k$ holds, then we have

$$
\begin{aligned}
f(\mathbf{x}_k) &\leq \min_{\mathbf{x}\in\mathcal{X}}\Phi_k(\mathbf{x}) + \xi_k \leq \Phi_k(\mathbf{x}^*) + \xi_k \\
&\leq (1-\lambda_k)f(\mathbf{x}^*) + \lambda_k\Phi_0(\mathbf{x}^*) + \xi_k
\end{aligned}
$$

where the last inequality is because Definition 5. Subtracting $f(\mathbf{x}^*)$ on both sides, we arrive at

$$
\begin{aligned}
f(\mathbf{x}_k) - f(\mathbf{x}^*) &\leq \lambda_k\big(\Phi_0(\mathbf{x}^*) - f(\mathbf{x}^*)\big) + \xi_k \\
&= \lambda_k\big(f(\mathbf{x}_0) - f(\mathbf{x}^*)\big) + \xi_k
\end{aligned}
$$

which completes the proof. □

Lemma 6. *Let* $\mathbf{v}_0 = \mathbf{x}_0$*,* $\boldsymbol{\theta}_0 = \mathbf{0}$*,* $\Phi_0^* = f(\mathbf{x}_0)$*, then* $\Phi_{k+1}(\mathbf{x})$ *in* (3.4) *can be rewritten as*

$$\Phi_{k+1}(\mathbf{x}) = \Phi_{k+1}^* + \langle \mathbf{x} - \mathbf{v}_{k+1}, \boldsymbol{\theta}_{k+1}\rangle \tag{3.13}$$

with

$$\boldsymbol{\theta}_{k+1} = \delta_k \nabla f(\mathbf{y}_k) + (1-\delta_k)\boldsymbol{\theta}_k \tag{3.14a}$$

$$\mathbf{v}_{k+1} := \arg\min_{\mathbf{x}\in\mathcal{X}} \Phi_{k+1}(\mathbf{x}) = \arg\min_{\mathbf{x}\in\mathcal{X}} \langle \mathbf{x}, \boldsymbol{\theta}_{k+1}\rangle \tag{3.14b}$$

$$\begin{aligned}\Phi^*_{k+1} &:= \min_{\mathbf{x}\in\mathcal{X}} \Phi_{k+1}(\mathbf{x}) = \Phi_{k+1}(\mathbf{v}_{k+1}) \\ &= (1-\delta_k)\Phi^*_k + \delta_k f(\mathbf{y}_k) + (1-\delta_k)\langle \boldsymbol{\theta}_k, \mathbf{v}_{k+1} - \mathbf{v}_k\rangle + \delta_k \langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{y}_k\rangle.\end{aligned} \tag{3.14c}$$

Proof. We prove this lemma by induction. First $\Phi_0(\mathbf{x}) = \Phi^*_0 + \langle \mathbf{x} - \mathbf{v}_0, \boldsymbol{\theta}_0\rangle \equiv f(\mathbf{x}_0)$. From (3.4) it is obvious that $\Phi_k(\mathbf{x})$ is linear in $\mathbf{x}$, and hence suppose that $\Phi_k(\mathbf{x}) = \Phi^*_k + \langle \mathbf{x} - \mathbf{v}_k, \boldsymbol{\theta}_k\rangle$ holds for some k. Then we will show that $\Phi_{k+1}(\mathbf{x}) = \Phi^*_{k+1} + \langle \mathbf{x} - \mathbf{v}_{k+1}, \boldsymbol{\theta}_{k+1}\rangle$ is true. Consider that

$$\begin{aligned}\Phi_{k+1}(\mathbf{x}) &= (1-\delta_k)\Phi_k(\mathbf{x}) + \delta_k \Big[f(\mathbf{y}_k) + \big\langle \nabla f(\mathbf{y}_k), \mathbf{x} - \mathbf{y}_k \big\rangle \Big] \\ &= (1-\delta_k)\Phi^*_k + (1-\delta_k)\langle \mathbf{x} - \mathbf{v}_k, \boldsymbol{\theta}_k\rangle + \delta_k f(\mathbf{y}_k) + \delta_k \big\langle \nabla f(\mathbf{y}_k), \mathbf{x} - \mathbf{y}_k\big\rangle \\ &= (1-\delta_k)\Phi^*_k + \delta_k f(\mathbf{y}_k) + \big\langle \mathbf{x}, (1-\delta_k)\boldsymbol{\theta}_k + \delta_k \nabla f(\mathbf{y}_k)\big\rangle \\ &\qquad - (1-\delta_k)\langle \mathbf{v}_k, \boldsymbol{\theta}_k\rangle - \delta_k \big\langle \nabla f(\mathbf{y}_k), \mathbf{y}_k\big\rangle.\end{aligned} \tag{3.15}$$

Clearly, since $\Phi_{k+1}(\mathbf{x})$ is linear in $\mathbf{x}$, the slope is $\boldsymbol{\theta}_{k+1} := (1-\delta_k)\boldsymbol{\theta}_k + \delta_k \nabla f(\mathbf{y}_k)$. In addition, because $\mathbf{v}_{k+1}$ is defined as the minimizer of $\Phi_{k+1}(\mathbf{x})$ over $\mathcal{X}$, from (3.15) we have $\mathbf{v}_{k+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \langle \mathbf{x}, \boldsymbol{\theta}_{k+1}\rangle$. Then, since Φ^*_{k+1} is defined as $\Phi^*_{k+1} := \min_{\mathbf{x}\in\mathcal{X}} \Phi_{k+1}(\mathbf{x})$, by plugging $\mathbf{v}_{k+1}$ into $\Phi_{k+1}(\mathbf{x})$ in (3.15), we have

$$\begin{aligned}\Phi^*_{k+1} = \Phi_{k+1}(\mathbf{v}_{k+1}) &= (1-\delta_k)\langle \mathbf{v}_{k+1} - \mathbf{v}_k, \boldsymbol{\theta}_k\rangle \\ &\quad + (1-\delta_k)\Phi^*_k + \delta_k f(\mathbf{y}_k) + \delta_k \big\langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{y}_k \big\rangle.\end{aligned}$$

The proof is thus completed. □

3.5.6 Proof of Lemma 4.

Proof. We prove this lemma by induction. First by definition $f(\mathbf{x}_0) = \Phi^*_0 + \xi_0$. Suppose now we have $f(\mathbf{x}_k) \leq \Phi^*_k + \xi_k$ for some k. Next, we will show that $f(\mathbf{x}_{k+1}) \leq \Phi^*_{k+1} + \xi_{k+1}$.

Using (3.14c), we have

$$
\begin{aligned}
&\Phi_{k+1}^* + (1-\delta_k)\xi_k \\
&= (1-\delta_k)\Phi_k^* + \delta_k f(\mathbf{y}_k) + (1-\delta_k)\langle \boldsymbol{\theta}_k, \mathbf{v}_{k+1} - \mathbf{v}_k \rangle \\
&\qquad + \delta_k \langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{y}_k \rangle + (1-\delta_k)\xi_k \\
&\overset{(a)}{\geq} (1-\delta_k) f(\mathbf{x}_k) + \delta_k f(\mathbf{y}_k) + (1-\delta_k)\langle \boldsymbol{\theta}_k, \mathbf{v}_{k+1} - \mathbf{v}_k \rangle \\
&\qquad + \delta_k \langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{y}_k \rangle \\
&\overset{(b)}{\geq} (1-\delta_k) f(\mathbf{x}_k) + \delta_k f(\mathbf{y}_k) + \delta_k \langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{y}_k \rangle \\
&= f(\mathbf{y}_k) + (1-\delta_k)\big[f(\mathbf{x}_k) - f(\mathbf{y}_k)\big] + \delta_k \langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{y}_k \rangle \\
&\overset{(c)}{\geq} f(\mathbf{y}_k) + (1-\delta_k)\big\langle \nabla f(\mathbf{y}_k), \mathbf{x}_k - \mathbf{y}_k \big\rangle + \delta_k \langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{y}_k \rangle \\
&\overset{(d)}{\geq} f(\mathbf{x}_{k+1}) - \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{y}_k\|^2 + \langle \nabla f(\mathbf{y}_k), \mathbf{y}_k - \mathbf{x}_{k+1} \rangle \\
&\qquad + (1-\delta_k)\big\langle \nabla f(\mathbf{y}_k), \mathbf{x}_k - \mathbf{y}_k \big\rangle + \delta_k \langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{y}_k \rangle \\
&\overset{(e)}{=} f(\mathbf{x}_{k+1}) - \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{y}_k\|^2
\end{aligned}
$$

where (a) is because $\Phi_k^* \geq f(\mathbf{x}_k) - \xi_k$; (b) is by the fact $\mathbf{v}_k = \arg\min_{\mathbf{x}\in\mathcal{X}} \langle \boldsymbol{\theta}_k, \mathbf{x} \rangle$ so that $\langle \boldsymbol{\theta}_k, \mathbf{v}_{k+1} - \mathbf{v}_k \rangle \geq 0$; (c) is because of the convexity of f; (d) is by Assumption 1, that is $f(\mathbf{x}_{k+1}) - f(\mathbf{y}_k) \leq \langle \nabla f(\mathbf{y}_k), \mathbf{x}_{k+1} - \mathbf{y}_k \rangle + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{y}_k\|^2$; (e) follows from the choice of $\mathbf{x}_{k+1} = (1-\delta_k)\mathbf{x}_k + \delta_k \mathbf{v}_{k+1}$. Finally by using $\mathbf{y}_k = (1-\delta_k)\mathbf{x}_k + \delta_k \mathbf{v}_k$, and plugging the definition of ξ_{k+1}, the proof is completed. □

3.5.7 Proof of Theorem 9

Proof. Since Lemma 4 holds, one can directly apply Lemma 5 to have

$$
\begin{aligned}
f(\mathbf{x}_k) - f(\mathbf{x}^*) &\leq \lambda_k\big(f(\mathbf{x}_0) - f(\mathbf{x}^*)\big) + \xi_k \\
&= \frac{2\big(f(\mathbf{x}_0) - f(\mathbf{x}^*)\big)}{(k+1)(k+2)} + \xi_k
\end{aligned} \tag{3.16}
$$

where ξ_k is defined in Lemma 4. Clearly, $\xi_k \geq 0,\ \forall k$, and we can find an upper bound for it in the following manner.

$$\begin{aligned}
\xi_k &= (1-\delta_{k-1})\xi_{k-1} + \frac{L\delta_{k-1}^2}{2}\|\mathbf{v}_k - \mathbf{v}_{k-1}\|^2 \\
&\leq (1-\delta_{k-1})\xi_{k-1} + \frac{LD^2\delta_{k-1}^2}{2} \\
&= \frac{LD^2}{2}\sum_{\tau=0}^{k-1}\delta_\tau^2\Big[\prod_{j=\tau+1}^{k-1}(1-\delta_j)\Big] \\
&= \frac{LD^2}{2}\sum_{\tau=0}^{k-1}\frac{4}{(\tau+3)^2}\frac{(\tau+2)(\tau+3)}{(k+1)(k+2)} \leq \frac{2LD^2}{k+2}.
\end{aligned}$$

Plugging ξ_k into (3.16) completes the proof. □

3.5.8 Preparation to the proof of Theorem 10

The basic idea is to show that under Assumptions 1, 2, 3 and 4, $\|\mathbf{v}_k - \mathbf{v}_{k+1}\|^2$ is small enough when k is large. To this end, we will make use of the following lemmas.

Lemma 7. *[8, Theorem 2.1.5] If Assumptions 1 and 2 hold, then it is true that*

$$\frac{1}{2L}\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|^2 \leq f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle.$$

Next we show that the value of $\nabla f(\mathbf{x}^*)$ is unique.

Lemma 8. *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 7, we have

$$\begin{aligned}
\frac{1}{2L}\|\nabla f(\mathbf{x}_2^*) - \nabla f(\mathbf{x}_1^*)\|_2^2 &\leq f(\mathbf{x}_2^*) - f(\mathbf{x}_1^*) - \langle \nabla f(\mathbf{x}_1^*), \mathbf{x}_2^* - \mathbf{x}_1^*\rangle \\
&\overset{(a)}{\leq} f(\mathbf{x}_2^*) - f(\mathbf{x}_1^*) = 0
\end{aligned}$$

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 9. *Choose $\delta_k = \frac{2}{k+3}$ and let $M := \max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) - f(\mathbf{x}^*)$, then we have*

$$\|\nabla f(\mathbf{y}_k) - \nabla f(\mathbf{x}^*)\| \leq \frac{C_1}{\sqrt{k+3}}.$$

where $C_1 = \sqrt{6LM + 4L^2D^2}$.

Proof. By convexity

$$\begin{aligned}
f(\mathbf{y}_k) - f(\mathbf{x}^*) &\leq (1-\delta_k)\big[f(\mathbf{x}_k) - f(\mathbf{x}^*)\big] + \delta_k\big[f(\mathbf{v}_k) - f(\mathbf{x}^*)\big] \\
&\overset{(a)}{\leq} \frac{k+1}{k+3}\left[\frac{2\big(f(\mathbf{x}_0) - f(\mathbf{x}^*)\big)}{(k+1)(k+2)} + \frac{2LD^2}{k+2}\right] + \frac{2M}{k+3} \\
&\leq \frac{2M}{(k+2)(k+3)} + \frac{2LD^2}{k+3} + \frac{2M}{k+3} \\
&\leq \frac{3M + 2LD^2}{k+3}
\end{aligned}$$

where (a) is by Theorem 9. Next using Lemma 7, we have

$$\begin{aligned}
\frac{1}{2L}\|\nabla f(\mathbf{y}_k) - \nabla f(\mathbf{x}^*)\|^2 &\leq f(\mathbf{y}_k) - f(\mathbf{x}^*) - \langle \nabla f(\mathbf{x}^*), \mathbf{y}_k - \mathbf{x}^*\rangle \\
&\overset{(b)}{\leq} f(\mathbf{y}_k) - f(\mathbf{x}^*) \leq \frac{3M + 2LD^2}{k+3}
\end{aligned}$$

where (b) is by the optimality condition, that is, $\langle \nabla f(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^*\rangle \geq 0,\ \forall \mathbf{x} \in \mathcal{X}$. This further implies

$$\|\nabla f(\mathbf{y}_k) - \nabla f(\mathbf{x}^*)\| \leq \sqrt{\frac{2L(3M + 2LD^2)}{k+3}}.$$

The proof is thus completed. □

Lemma 10. *Choose $\delta_k = \frac{2}{k+3}$, it is guaranteed to have*

$$\|\boldsymbol{\theta}_{k+1} - \nabla f(\mathbf{x}^*)\| \leq \frac{4C_1}{3(\sqrt{k+3}-1)} + \frac{2\sqrt{G}}{(k+2)(k+3)}.$$

In addition, there exists a constant $C_2 \leq \frac{4}{3}C_1 + \frac{2}{3(\sqrt{3}+1)}\sqrt{G}$ such that

$$\|\boldsymbol{\theta}_{k+1} - \nabla f(\mathbf{x}^*)\| \leq \frac{C_2}{\sqrt{k+3}-1}.$$

Proof. First we have

$$\begin{aligned}
\boldsymbol{\theta}_{k+1} &= (1-\delta_k)\boldsymbol{\theta}_k + \delta_k \nabla f(\mathbf{y}_k) \qquad (3.17) \\
&= \sum_{\tau=0}^{k} \delta_\tau \nabla f(\mathbf{y}_\tau)\left[\prod_{j=\tau+1}^{k}(1-\delta_j)\right] \\
&= \sum_{\tau=0}^{k} \frac{2(\tau+2)}{(k+2)(k+3)} \nabla f(\mathbf{y}_\tau).
\end{aligned}$$

Noticing that $2\sum_{\tau=0}^{k}(\tau+2) = (k+1)(k+4) = (k+2)(k+3) - 2$, we have

$$
\begin{aligned}
&\|\boldsymbol{\theta}_{k+1} - \nabla f(\mathbf{x}^*)\| \\
&= \Big\| \sum_{\tau=0}^{k} \frac{2(\tau+2)}{(k+2)(k+3)} \big[\nabla f(\mathbf{y}_\tau) - \nabla f(\mathbf{x}^*)\big] - \frac{2}{(k+2)(k+3)} \nabla f(\mathbf{x}^*) \Big\| \\
&\leq \sum_{\tau=0}^{k} \frac{2(\tau+2)}{(k+2)(k+3)} \big\|\nabla f(\mathbf{y}_\tau) - \nabla f(\mathbf{x}^*)\big\| + \frac{2}{(k+2)(k+3)} \big\|\nabla f(\mathbf{x}^*)\big\| \\
&\overset{(a)}{\leq} \sum_{\tau=0}^{k} \frac{2(\tau+2)}{(k+2)(k+3)} \frac{C_1}{\sqrt{\tau+3}} + \frac{2\sqrt{G}}{(k+2)(k+3)} \\
&\leq \frac{2C_1}{(k+2)(k+3)} \sum_{\tau=0}^{k} \sqrt{\tau+2} + \frac{2\sqrt{G}}{(k+2)(k+3)} \\
&\leq \frac{4C_1}{3(k+2)(k+3)} (k+3)^{3/2} + \frac{2\sqrt{G}}{(k+2)(k+3)} \\
&= \frac{4C_1}{3(\sqrt{k+3}+1)(\sqrt{k+3}-1)} \sqrt{k+3} + \frac{2\sqrt{G}}{(k+2)(k+3)} \\
&\leq \frac{4C_1}{3(\sqrt{k+3}-1)} + \frac{2\sqrt{G}}{(k+2)(k+3)}
\end{aligned}
$$

where (a) follows from Lemma 9 and Assumption 4.

Then to find C_2, we have

$$
\begin{aligned}
&\|\boldsymbol{\theta}_{k+1} - \nabla f(\mathbf{x}^*)\| \\
&\leq \frac{4C_1}{3(\sqrt{k+3}-1)} + \frac{2\sqrt{G}}{(k+2)(k+3)} \\
&= \frac{4C_1}{3(\sqrt{k+3}-1)} + \frac{2\sqrt{G}}{(k+3)(\sqrt{k+3}+1)(\sqrt{k+3}-1)} \\
&\overset{(b)}{\leq} \frac{4C_1}{3(\sqrt{k+3}-1)} + \frac{2\sqrt{G}}{3(\sqrt{3}+1)(\sqrt{k+3}-1)}
\end{aligned}
$$

where in (b) we use $k+3 \geq 3$ and $\sqrt{k+3}+1 \geq \sqrt{3}+1$. The proof is thus completed. □

Lemma 11. *There exists a constant $T \leq \big(\frac{2C_2}{\sqrt{G}}+1\big)^2 - 3$, such that $\|\boldsymbol{\theta}_{k+1}\| \geq \frac{\sqrt{G}}{2}$, $\forall k \geq T$. In addition, it is guaranteed to have for any $k \geq T+1$*

$$\|\mathbf{v}_{k+1} - \mathbf{v}_k\| \leq \frac{C_3}{\sqrt{k+2}-1}$$

where $C_3 \leq \frac{4R}{G}\big[4\sqrt{G}C_2 + \frac{2C_2^2}{\sqrt{T+4}-1}\big]$.

Proof. Consider a specific $\tilde{k}$ with $\|\boldsymbol{\theta}_{\tilde{k}+1}\| < \frac{\sqrt{G}}{2}$ satisfied. In this case we have

$$\|\boldsymbol{\theta}_{\tilde{k}+1} - \nabla f(\mathbf{x}^*)\| \geq \|\nabla f(\mathbf{x}^*)\| - \|\boldsymbol{\theta}_{\tilde{k}+1}\| > \sqrt{G} - \frac{\sqrt{G}}{2} = \frac{\sqrt{G}}{2}.$$

From Lemma 10, we have

$$\frac{\sqrt{G}}{2} < \|\boldsymbol{\theta}_{\tilde{k}+1} - \nabla f(\mathbf{x}^*)\| \leq \frac{C_2}{\sqrt{\tilde{k}+3}-1}.$$

From this inequality we can observe that $\|\boldsymbol{\theta}_{\tilde{k}+1}\|$ can be less than $\frac{\sqrt{G}}{2}$ only when $\tilde{k} < T = \big(\frac{2C_2}{\sqrt{G}}+1\big)^2 - 3$. Hence, the first part of this lemma is proved.

For the upper bound of $\|\mathbf{v}_{k+1} - \mathbf{v}_k\|$, we only consider the case where $\boldsymbol{\theta}_{k+1} \neq \mathbf{0}$ since otherwise $\mathbf{v}_{k+1} = \mathbf{v}_k$ and the lemma holds automatically. For any $k \geq T+1$, from (3.5), one can rewrite

$$\begin{aligned}
&\|\mathbf{v}_{k+1} - \mathbf{v}_k\| \\
&= R\bigg\|\frac{\boldsymbol{\theta}_{k+1}}{\|\boldsymbol{\theta}_{k+1}\|} - \frac{\boldsymbol{\theta}_k}{\|\boldsymbol{\theta}_k\|}\bigg\| \\
&= \frac{R}{\|\boldsymbol{\theta}_{k+1}\|\|\boldsymbol{\theta}_k\|}\Big\|\|\boldsymbol{\theta}_k\|\boldsymbol{\theta}_{k+1} - \|\boldsymbol{\theta}_{k+1}\|\boldsymbol{\theta}_k\Big\| \\
&\overset{(a)}{\leq} \frac{4R}{G}\Big\|\|\boldsymbol{\theta}_k\|\boldsymbol{\theta}_{k+1} - \|\boldsymbol{\theta}_{k+1}\|\boldsymbol{\theta}_k\Big\|
\end{aligned} \tag{3.18}$$

where (a) is by $\boldsymbol{\theta}_k \geq \frac{\sqrt{G}}{2}$ for $k \geq T+1$. Next we rewrite $\boldsymbol{\theta}_k := \nabla f(\mathbf{x}^*) + \boldsymbol{\gamma}_k$. From Lemma 10 we have $\|\boldsymbol{\gamma}_k\| = \|\boldsymbol{\theta}_k - \nabla f(\mathbf{x}^*)\| \leq \frac{C_2}{\sqrt{k+2}-1}$. Using this relation, the RHS of (3.18) becomes

$$\begin{aligned}
&\Big\|\|\boldsymbol{\theta}_k\|\boldsymbol{\theta}_{k+1} - \|\boldsymbol{\theta}_{k+1}\|\boldsymbol{\theta}_k\Big\| \\
&= \Big\|\big\|\nabla f(\mathbf{x}^*) + \boldsymbol{\gamma}_k\big\|\big(\nabla f(\mathbf{x}^*) + \boldsymbol{\gamma}_{k+1}\big) - \big\|\nabla f(\mathbf{x}^*) + \boldsymbol{\gamma}_{k+1}\big\|\big(\nabla f(\mathbf{x}^*) + \boldsymbol{\gamma}_k\big)\Big\| \\
&\leq \|\nabla f(\mathbf{x}^*)\|\Big|\big\|\nabla f(\mathbf{x}^*) + \boldsymbol{\gamma}_k\big\| - \big\|\nabla f(\mathbf{x}^*) + \boldsymbol{\gamma}_{k+1}\big\|\Big| \\
&\quad + \Big\|\boldsymbol{\gamma}_{k+1}\big\|\nabla f(\mathbf{x}^*) + \boldsymbol{\gamma}_k\big\| - \boldsymbol{\gamma}_k\big\|\nabla f(\mathbf{x}^*) + \boldsymbol{\gamma}_{k+1}\big\|\Big\| \\
&\leq \sqrt{G}\big(\|\boldsymbol{\gamma}_k\| + \|\boldsymbol{\gamma}_{k+1}\|\big) + \|\boldsymbol{\gamma}_{k+1}\|\big(\sqrt{G} + \|\boldsymbol{\gamma}_k\|\big) + \|\boldsymbol{\gamma}_k\|\big(\sqrt{G} + \|\boldsymbol{\gamma}_{k+1}\|\big) \\
&\leq \frac{4\sqrt{G}C_2}{\sqrt{k+2}-1} + \frac{2C_2^2}{(\sqrt{k+2}-1)(\sqrt{k+3}-1)} \\
&\leq \frac{4\sqrt{G}C_2}{\sqrt{k+2}-1} + \frac{2C_2^2}{(\sqrt{k+2}-1)(\sqrt{T+4}-1)}.
\end{aligned}$$

Plugging back to (3.18), the proof can be completed. □

3.5.9 Proof of Theorem 10.

Proof. We first consider the constraint set being an ℓ_2 norm ball. From Lemma 4, we can write

$$\begin{aligned}
\xi_{k+1} &= (1-\delta_k)\xi_k + \frac{L\delta_k^2}{2}\|\mathbf{v}_{k+1}-\mathbf{v}_k\|^2 \\
&= \frac{L}{2}\sum_{\tau=0}^{k}\delta_\tau^2\|\mathbf{v}_{\tau+1}-\mathbf{v}_\tau\|^2\bigg[\prod_{j=\tau+1}^{k}(1-\delta_\tau)\bigg] \\
&\overset{(a)}{=} \frac{L}{2}\sum_{\tau=0}^{T}\delta_\tau^2\|\mathbf{v}_{\tau+1}-\mathbf{v}_\tau\|^2\bigg[\prod_{j=\tau+1}^{k}(1-\delta_\tau)\bigg] + \sum_{\tau=T+1}^{k}\delta_\tau^2\|\mathbf{v}_{\tau+1}-\mathbf{v}_\tau\|^2\bigg[\prod_{j=\tau+1}^{k}(1-\delta_\tau)\bigg] \\
&\overset{(b)}{\leq} \frac{L}{2}\sum_{\tau=0}^{T}\delta_\tau^2 D^2\bigg[\prod_{j=\tau+1}^{k}(1-\delta_\tau)\bigg] + \sum_{\tau=T+1}^{k}\delta_\tau^2\frac{C_3^2}{(\sqrt{\tau+2}-1)^2}\bigg[\prod_{j=\tau+1}^{k}(1-\delta_\tau)\bigg] \\
&= \frac{L}{2}\sum_{\tau=0}^{T}\frac{4D^2}{(\tau+3)^2}\frac{(\tau+2)(\tau+3)}{(k+2)(k+3)} + \sum_{\tau=T+1}^{k}\frac{4}{(\tau+3)^2}\frac{C_3^2}{(\sqrt{\tau+2}-1)^2}\frac{(\tau+2)(\tau+3)}{(k+2)(k+3)} \\
&\leq \frac{2LD^2(T+1)}{(k+2)(k+3)} + \frac{4C_3^2}{(k+2)(k+3)}\sum_{\tau=T+1}^{k}\frac{1}{(\sqrt{\tau+2}-1)^2} \\
&= \mathcal{O}\bigg(\frac{LD^2(T+1)+C_3^2\ln k}{(k+2)(k+3)}\bigg)
\end{aligned}$$

where in (a) T is defined in Lemma 11; (b) is by Lemma 11 and Assumption 4; and in the last equation constants are hide in the big $\mathcal{O}$ notation.

Finally, applying Lemma 5, we have

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{2\big[f(\mathbf{x}_0)-f(\mathbf{x}^*)\big]}{(k+1)(k+2)} + \xi_k. \tag{3.19}$$

Plugging ξ_k in the proof is completed.

□

3.5.10 ℓ_1 norm ball

In this subsection we focus on the convergence of AFW for ℓ_1 norm ball constraint under the assumption that $\arg\max_j \big|[\nabla f(\mathbf{x}^*)]_j\big|$ has cardinality 1 (which naturally implies that the constraint is active). Note that in this case Lemma 8 still holds hence the value of $\nabla f(\mathbf{x}^*)$ is unique regardless the uniqueness of $\mathbf{x}^*$. This assumption directly leads to $\arg\max_j \big|[\nabla f(\mathbf{x}^*)]_j\big| - |[\nabla f(\mathbf{x}^*)]_i| \geq \lambda,\ \forall i$.

When $\mathcal{X} = \{\mathbf{x} | \|\mathbf{x}\|_1 \leq R\}$, the FW steps for AFW can be solved in closed-form. We have $\mathbf{v}_{k+1} = [0, \ldots, 0, -\text{sgn}[\boldsymbol{\theta}_{k+1}]_i R, 0, \ldots, 0]^\top$, i.e., only the i-th entry being nonzero with $i = \arg\max_j |[\boldsymbol{\theta}_{k+1}]_j|$.

Lemma 12. *There exist a constant T (which is irreverent with k), whenever $k \geq T$, it is guaranteed to have*

$$\|\mathbf{v}_{k+1} - \mathbf{v}_{k+2}\| = 0$$

Proof. In the proof, we denote $i = \arg\max_j |[\nabla f(\mathbf{x}^*)]_j|$ for convenience. It can be seen that Lemma 10 still holds.

We show that there exist $T = (\frac{3C_2}{\lambda}+1)^2-3$, such that for all $k \geq T$, we have $\arg\max_j |[\boldsymbol{\theta}_{k+1}]_j| = i$, which further implies only the i-th entry of $\mathbf{v}_{k+1}$ is non-zero. Since Lemma 10 holds, one can see whenever $k \geq T$, it is guaranteed to have $\|\boldsymbol{\theta}_{k+1} - \nabla f(\mathbf{x}^*)\| \leq \frac{\lambda}{3}$. Therefore, one must have $\big| |[\boldsymbol{\theta}_{k+1}]_j| - |[\nabla f(\mathbf{x}^*)]_j| \big| \leq \frac{\lambda}{3}$, $\forall j$. Then it is easy to see that $|[\boldsymbol{\theta}_{k+1}]_i| - |[\boldsymbol{\theta}_{k+1}]_j| \geq \frac{\lambda}{3}$, $\forall j$. Hence, we have $\arg\max_j |[\boldsymbol{\theta}_{k+1}]_j| = i$.

Then one can use the closed form solution of FW step to see that when $k \geq T$, we have $\mathbf{v}_{k+1} - \mathbf{v}_{k+2} = \mathbf{0}$. The proof is thus completed. □

Lemma 13. *Let $\xi_0 = 0$ and T defined the same as in Lemma 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 $k < T+1$, $\xi_{k+1} = (1-\delta_k)\xi_k + \frac{LD^2}{2}\delta_k^2$, and $\xi_{k+1} = (1-\delta_k)\xi_k$ for $k \geq T+1$.

Proof. The proof for $k < T+1$ is similar as that in Lemma 4, hence it is omitted here. For $k \geq T+1$, using similar argument as in Lemma 4, we have

$$\begin{aligned}\Phi_{k+1}^* &\geq f(\mathbf{x}_{k+1}) + \frac{L\delta_k^2}{2}\|\mathbf{v}_{k+1} - \mathbf{v}_k\|^2 - (1-\delta_k)\xi_k \\ &= f(\mathbf{x}_{k+1}) - (1-\delta_k)\xi_k\end{aligned}$$

where the last equation is because of Lemma 12. □

Theorem 12. *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 1 - 3 are satisfied, AFW guarantees 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 25. For convenience denote $\xi_{k+1} = (1-\delta_k)\xi_k + \zeta_k$. When $k < T+1$, we have $\zeta_k = \frac{LD^2}{2}\delta_k^2$; when $k \geq T+1$, we have $\zeta_k = 0$. Then we can write

$$\begin{aligned}\xi_{k+1} &= (1-\delta_k)\xi_k + \theta_k \\ &= \sum_{\tau=0}^{k} \theta_\tau \prod_{j=\tau+1}^{k} (1-\delta_j) = \sum_{\tau=0}^{k} \theta_\tau \frac{(\tau+2)(\tau+3)}{(k+2)(k+3)} \\ &= \sum_{\tau=0}^{T} \frac{LD^2}{2} \delta_\tau^2 \frac{(\tau+2)(\tau+3)}{(k+2)(k+3)} = \frac{2LD^2(T+1)}{(k+2)(k+3)}.\end{aligned}$$

Finally, applying Lemma 5, we have

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{2\big[f(\mathbf{x}_0) - f(\mathbf{x}^*)\big]}{(k+1)(k+2)} + \xi_k.$$

Plugging ξ_k in completes the proof. □

3.5.11 ℓ_p norm ball

In this subsection we focus on AFW with an active ℓ_p norm ball constraint $\mathcal{X} := \{\mathbf{x} | \|\mathbf{x}\|_p \leq R\}$, where $p \in (1, +\infty)$ and $p \neq 2$. We show that if the magnitude of every entry in $\nabla f(\mathbf{x}^*)$ is bounded away from 0, i.e., $|[\nabla f(\mathbf{x}^*)]_i| = \lambda > 0, \ \forall i$, then AFW converges at $\mathcal{O}(\frac{1}{k^2})$.

In such cases, the FW step in AFW can be solved in closed-form, that is, the i-th entry of $\mathbf{v}_{k+1}$ can be obtained via

$$\begin{aligned}[\mathbf{v}_{k+1}]_i &= -\mathrm{sgn}\big([\boldsymbol{\theta}_{k+1}]_i\big) \frac{\big|[\boldsymbol{\theta}_{k+1}]_i\big|^{q-1}}{\|\boldsymbol{\theta}_{k+1}\|_q^{q-1}} \cdot R \\ &= -[\boldsymbol{\theta}_{k+1}]_i \frac{\big|[\boldsymbol{\theta}_{k+1}]_i\big|^{q-2}}{\|\boldsymbol{\theta}_{k+1}\|_q^{q-1}} \cdot R\end{aligned} \tag{3.20}$$

where $1/p + 1/q = 1$. For simplicity we will emphasis on the k dependence only and use $\mathcal{O}$ notation in this subsection. We will also use θ_k^i to replace $[\boldsymbol{\theta}_k]_i$ for notational simplicity. In other words, θ_k^i denotes the i-th entry of $\boldsymbol{\theta}_k$.

First according to Lemma 10, and use the equivalence of norms, we have $\|\boldsymbol{\theta}_k - \nabla f(\mathbf{x}^*)\|_q = \mathcal{O}(\frac{1}{\sqrt{k}})$. Hence, there must exist T_1, such that $\|\boldsymbol{\theta}_k\|_q \leq 2G, \ \forall k \geq T_1$. Next using similar arguments as the first part of Lemma 11, there must exist T_2, such that $\|\boldsymbol{\theta}_k\|_q \geq G/2, \ \forall k \geq T_2$.

In addition, using again similar arguments as the first part of Lemma 11, we can find that there exist T_3, such that $|\theta_k^i| > \frac{\lambda}{2},\ \forall k \geq T_3$.

Let $T := \max\{T_1, T_2, T_3\}$. Next we will show that $\|\mathbf{v}_{k+1} - \mathbf{v}_k\|^2 = \mathcal{O}(\frac{1}{k}),\ \forall k \geq T$. To start, using (3.20), one can have

$$\begin{aligned}
& v_{k+1}^i - v_k^i \\
&= \frac{R}{\|\boldsymbol{\theta}_{k+1}\|_q^{q-1}\|\boldsymbol{\theta}_k\|_q^{q-1}}\Big[-\theta_{k+1}^i|\theta_{k+1}^i|^{q-2}\|\boldsymbol{\theta}_k\|_q^{q-1} + \theta_k^i|\theta_k^i|^{q-2}\|\boldsymbol{\theta}_{k+1}\|_q^{q-1}\Big] \\
&= \frac{R}{\|\boldsymbol{\theta}_{k+1}\|_q^{q-1}\|\boldsymbol{\theta}_k\|_q^{q-1}}\Big[\theta_{k+1}^i|\theta_{k+1}^i|^{q-2}\Big(\|\boldsymbol{\theta}_{k+1}\|_q^{q-1} - \|\boldsymbol{\theta}_k\|_q^{q-1}\Big) \\
&\qquad + \|\boldsymbol{\theta}_{k+1}\|_q^{q-1}\Big(\theta_k^i|\theta_k^i|^{q-2} - \theta_{k+1}^i|\theta_{k+1}^i|^{q-2}\Big)\Big].
\end{aligned}$$

Next using $G/2 \leq \|\boldsymbol{\theta}_{k+1}\|_q \leq 2G,\ \forall k \geq T$, and $|\theta_{k+1}^i| \leq \|\boldsymbol{\theta}_{k+1}\|_q$, we have

$$\begin{aligned}
& |v_{k+1}^i - v_k^i| \\
&= \mathcal{O}\Bigg(\Big|\|\boldsymbol{\theta}_{k+1}\|_q^{q-1} - \|\boldsymbol{\theta}_k\|_q^{q-1}\Big| + \Big|\theta_k^i|\theta_k^i|^{q-2} - \theta_{k+1}^i|\theta_{k+1}^i|^{q-2}\Big|\Bigg).
\end{aligned} \tag{3.21}$$

We first bound the first term in RHS of (3.21). Let $h(x) = (x)^{q-1}$. Then by mean value theorem we have $h(y) = h(x) + \nabla h(x)(y-x) + \nabla^2 h(z)\|x-y\|^2$, where $z = (1-\alpha)x + \alpha y$ for some $\alpha \in [0,1]$. Taking $x = \|\boldsymbol{\theta}_k\|_q$ and $y = \|\boldsymbol{\theta}_{k+1}\|_q$, and using the fact $G/2 \leq \|\boldsymbol{\theta}_k\|_q \leq 2G$ for $k \geq T$, we have

$$\begin{aligned}
& \|\boldsymbol{\theta}_{k+1}\|_q^{q-1} \\
&= \|\boldsymbol{\theta}_k\|_q^{q-1} + \mathcal{O}(\big|\|\boldsymbol{\theta}_k\|_q - \|\boldsymbol{\theta}_{k+1}\|_q\big| + \big|\|\boldsymbol{\theta}_k\|_q - \|\boldsymbol{\theta}_{k+1}\|_q\big|^2) \\
&= \|\boldsymbol{\theta}_k\|_q^{q-1} + \mathcal{O}(\frac{1}{\sqrt{k}})
\end{aligned} \tag{3.22}$$

Hence, one can find that the first term on the RHS of (3.21) is bounded by $\mathcal{O}(\frac{1}{\sqrt{k}})$.

Next we focus on the second term of (3.21) by considering whether θ_k^i and θ_{k+1}^i have different signs.

Case 1: θ_k^i and θ_{k+1}^i have the same sign. Then we have

$$\begin{aligned}
& \Big|\theta_k^i|\theta_k^i|^{q-2} - \theta_{k+1}^i|\theta_{k+1}^i|^{q-2}\Big| \\
&= \Big||\theta_k^i|^{q-1} - |\theta_{k+1}^i|^{q-1}\Big| \leq \mathcal{O}(\frac{1}{\sqrt{k}})
\end{aligned} \tag{3.23}$$

where the last inequality uses the same mean-value-theorem argument as (3.22) and the fact $|\theta_k^i| \geq \frac{\lambda}{2}$.

Case 2: θ_k^i and θ_{k+1}^i have different signs. We assume $\theta_{k+1}^i \geq 0$ w.l.o.g. In this case, by the update manner of $\boldsymbol{\theta}_{k+1}$, we have $|\theta_{k+1}^i| \leq |\delta_k [\nabla f(\mathbf{y}_k)]_i| = \mathcal{O}(\delta_k) = \mathcal{O}(\frac{1}{k})$. This is impossible given the fact $|\theta_{k+1}^i| > \frac{\lambda}{2}$ when $k \geq T$.

Therefore, we have the second term in (3.21) bounded by $\mathcal{O}(\frac{1}{\sqrt{k}})$. Hence, it is easy to see that

$$\|\mathbf{v}_{k+1} - \mathbf{v}_k\|^2 = \mathcal{O}\Big(\frac{1}{k}\Big).$$

Applying the same argument in the proof of Theorem 10, we have that when $k \geq T$, $\xi_{k+1} = \tilde{\mathcal{O}}(\frac{1}{k^2})$. This further implies $f(\mathbf{x}_k) - f(\mathbf{x}^*) = \tilde{\mathcal{O}}(\frac{1}{k^2})$ as well.

Chapter 4

Enhancing Parameter-Free Frank Wolfe with an Extra Subproblem

4.1 Introduction

In last chapter, we have discussed AFW, which replaces the subproblem of NAG by a single FW subproblem, and developed constraint-specific faster rates. 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, the present chapter 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 [88] and Mirror-Prox [83, 89, 90], leveraging PC updates in FW type algorithms for faster rates is novel.

Our contributions are summarized as follows.

- A new parameter-free FW variant, ExtraFW, is studied in this work. The distinct feature of ExtraFW is the adoption of two gradient evaluations per iteration to update the decision variable in a prediction-correction (PC) manner.

- It is shown that ExtraFW convergences with a rate of $\mathcal{O}(\frac{1}{k})$ for general problems. And for constraint sets including active ℓ_1, ℓ_2 and n-support norm balls, ExtraFW guarantees an accelerated rate $\mathcal{O}(\frac{1}{k^2})$.

- Unlike most of faster rates in FW literatures, ExtraFW is parameter-free, so that no problem dependent parameter is required. Compared with another parameter-free algorithm with faster rates, AFW [58], introducing PC update in ExtraFW leads to several advantages: i) the convergence rate is improved by a factor of $\mathcal{O}(\ln k)$ on an ℓ_2 norm ball constraint; and ii) the analysis does not rely on the maximum value of $f(\mathbf{x})$ over $\mathcal{X}$.
- The efficiency of ExtraFW is corroborated on two benchmark machine learning tasks. The faster rate $\mathcal{O}(\frac{1}{k^2})$ is achieved on binary classification, evidenced by the possible improvement of ExtraFW over NAG on multiple sparsity-promoting constraint sets. For matrix completion, ExtraFW improves over AFW and FW in both optimality error and the rank of the solution.

4.2 Preliminaries

This section reviews AFW in order to illustrate the proposed algorithm in a principled manner. We first pinpoint the class of problems to focus on.

AFW recap. As an FW variant, AFW in Alg. 5 relies on Nesterov's 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

$$\sum_{\tau=0}^{k} w_k^\tau \big[f(\mathbf{y}_\tau) + \langle \nabla f(\mathbf{y}_\tau), \mathbf{x} - \mathbf{y}_\tau \rangle\big] \tag{4.1}$$

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 (4.1) 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 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 (4.1) 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.

4.3 ExtraFW

This section introduces the main algorithm, ExtraFW, and establishes its constraint dependent faster rates.

4.3.1 Algorithm design

Algorithm 6 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: $\mathbf{x}_K$
```

ExtraFW is summarized in Alg. 6. Different from the vanilla FW and AFW, two FW steps (Lines 5 and 8 of Alg. 6) are required per iteration. Compared with other algorithms relying on two gradient evaluations, such as Mirror-Prox [89, 90], 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. To facilitate comparison with 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 Alg. 6 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,

$$\sum_{\tau=0}^{k-1} w_k^\tau \Big[f(\mathbf{x}_{\tau+1}) + \langle \nabla f(\mathbf{x}_{\tau+1}), \mathbf{x} - \mathbf{x}_{\tau+1} \rangle \Big] + \delta_k \Big[f(\mathbf{y}_k) + \big\langle \nabla f(\mathbf{y}_k), \mathbf{x} - \mathbf{y}_k \big\rangle \Big], \tag{4.2}$$

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 (4.2) 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 (4.2) 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 $f(\mathbf{x})$

$$\sum_{\tau=0}^{k-1} w_k^\tau \Big[f(\mathbf{x}_{\tau+1}) + \big\langle \nabla f(\mathbf{x}_{\tau+1}), \mathbf{x} - \mathbf{x}_{\tau+1} \big\rangle \Big] + \delta_k \Big[f(\mathbf{x}_{k+1}) + \big\langle \nabla f(\mathbf{x}_{k+1}), \mathbf{x} - \mathbf{x}_{k+1} \big\rangle \Big]. \tag{4.3}$$

Comparing (4.2) and (4.3), we deduce that the terms in the last bracket of (4.2) 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 (4.3). In the following subsections, 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.

4.3.2 Convergence of ExtraFW

We investigate the convergence of ExtraFW by considering the general case first. The analysis relies on the notion of estimate sequence (ES) introduced in [8] or Definition 5.

The construction of ES varies for different algorithms (see e.g., [80, 8, 76, 81]). However, the reason to rely on the ES based analysis is similar, as summarized in the following lemma.

Lemma 14. *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 14, 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 14 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

$$\Phi_0(\mathbf{x}) = \hat{\Phi}_0(\mathbf{x}) \equiv f(\mathbf{x}_0) \tag{4.4a}$$

$$\hat{\Phi}_{k+1}(\mathbf{x}) = (1-\delta_k)\Phi_k(\mathbf{x}) + \delta_k\big[f(\mathbf{y}_k) + \langle \nabla f(\mathbf{y}_k), \mathbf{x} - \mathbf{y}_k\rangle\big], \ \forall k \geq 0 \tag{4.4b}$$

$$\Phi_{k+1}(\mathbf{x}) = (1-\delta_k)\Phi_k(\mathbf{x}) + \langle \nabla f(\mathbf{x}_{k+1}), \mathbf{x} - \mathbf{x}_{k+1}\rangle\big], \ \forall k \geq 0. \tag{4.4c}$$

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 [8]. Such linear surrogate functions are constructed specifically for FW type algorithms taking advantage of the compact and convex constraint set. Next we show that (4.4) and proper $\{\lambda_k\}$ form two different ES of f.

Lemma 15. *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 (4.4) is that they are closely linked with the lower bounds (4.2) and (4.3) used in the FW steps, as stated in the next lemma.

Lemma 16. *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 (4.4) 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 14.

Lemma 17. *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 16), 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 17, 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 Appendix 4.5.6 due to space limitation. Now we are ready to apply Lemma 14 to establish the convergence of ExtraFW.

Theorem 13. *Suppose that Assumptions 1, 2 and 3 are satisfied. Choosing $\delta_k = \frac{2}{k+3}$, and $\mathbf{g}_0 = \mathbf{0}$, ExtraFW in Alg. 6 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 13 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 [3, 2].

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 13. 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.

4.3.3 Acceleration of ExtraFW

In this subsection, 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 [83, 89]. Thus, faster rates for parameter-free algorithms are challenging to establish. Similar to AFW, we also include Assumption 4 to enable faster convergence rate. More specifically, we assume that The constraint is active, i.e., $\|\nabla f(\mathbf{x}^*)\|_2 \geq G > 0$.

Technically, the need behind Assumption 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 test. Detailed analysis for ℓ_1 and n-support norm ball [12] constraints are provided in Appendix.

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,

$$\mathbf{v}_{k+1} = \arg\min_{\mathbf{x} \in \mathcal{X}} \langle \mathbf{g}_{k+1}, \mathbf{x} \rangle = -\frac{D}{2\|\mathbf{g}_{k+1}\|_2} \mathbf{g}_{k+1}. \tag{4.5}$$

We assume that when using $\mathbf{g}_{k+1}$ as the input to the LMO, the returned vector is given by (4.5). This is reasonable since it is what we usually implemented in practice. Though rarely happen, 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 14. *Suppose that Assumptions 1, 2, 3 and 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 Alg. 6 guarantees*

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) = \mathcal{O}\bigg(\frac{LD^2}{k} \wedge \frac{LD^2T}{k^2} \bigg), \forall k$$

where T is a constant depending only on L, G, and D.

Theorem 14 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 13 so that the parameter-free implementation is the same regardless whether accelerated. In other words, prior knowledge on whether Assumption 4 holds is not needed in practice. Compared with CGS, ExtraFW sacrifices the D dependence in the convergence rate to trade for i) the nonnecessity 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 [58], 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

$$\mathbf{v}_{k+1} = R \cdot [0, \ldots, 0, -\text{sgn}[\mathbf{g}_{k+1}]_i, 0, \ldots, 0]^\top \text{ with } i = \arg\max_j |[\mathbf{g}_{k+1}]_j|. \tag{4.6}$$

We show in Theorem 15 (see Appendix 4.5.8) that when Assumption 4 holds and the set $\arg\max_j \big|[\nabla f(\mathbf{x}^*)]_j\big|$ 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.,[85, 86].

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 [61]. 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 [12]. The closed-form solution of $\mathbf{v}_{k+1}$ is given by [62]

$$\mathbf{v}_{k+1} = -\frac{R}{\|\text{top}_n(\mathbf{g}_{k+1})\|_2}\text{top}_n(\mathbf{g}_{k+1}) \tag{4.7}$$

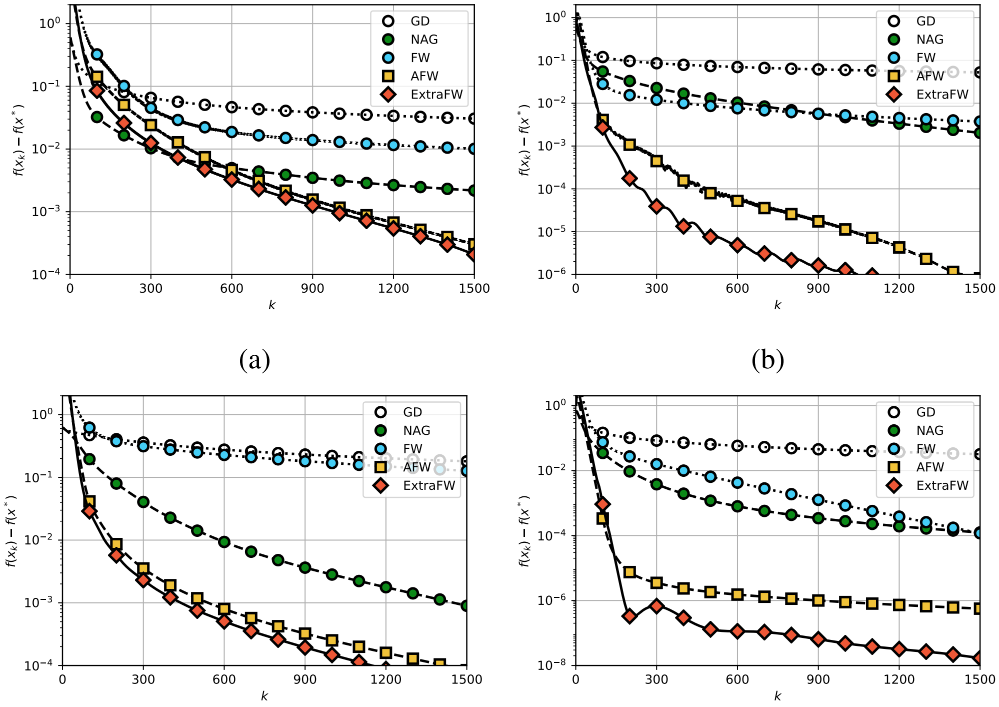
where $\text{top}_n(\mathbf{g})$ denotes the truncated version of $\mathbf{g}$ with its top n (in magnitude) entries preserved. A faster rate $\mathcal{O}(\frac{T_2 L D^2}{k^2})$ is guaranteed by ExtraFW under Assumption 4, and a condition similar to strict complementarity (see Theorem 16 in the Appendix 4.5.9). 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 Appendix 4.5.8. 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.

4.4 Numerical tests

This section deals with numerical tests 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 Appendix 4.5.10. For comparison, the benchmarked algorithms are chosen as: i) GD with standard step size $\frac{1}{L}$; ii) Nesterov accelerated gradient (NAG) with step sizes in [73]; iii) FW with parameter-free step size $\frac{2}{k+2}$ [2]; and iv) AFW with step size $\frac{2}{k+3}$ [58].

4.4.1 Binary classification



(a)

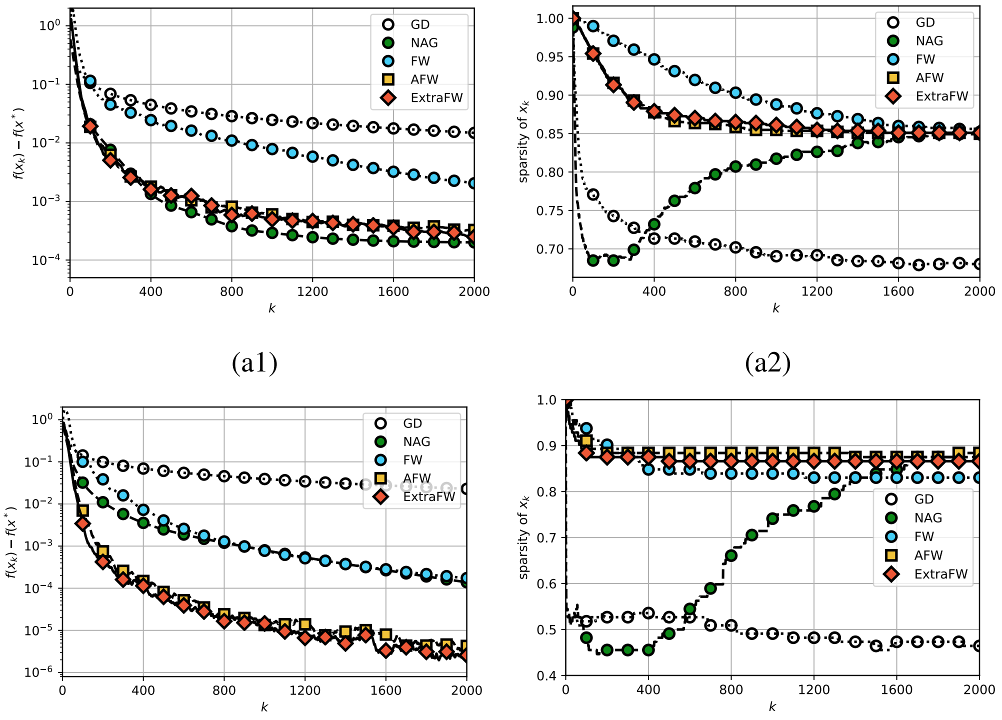
(b)

(c)

(d)

Figure 4.1: 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 error are plotted in Figure 4.1. On all tested datasets, ExtraFW outperforms AFW, NAG, FW and GD, demonstrating the $\mathcal{O}(\frac{1}{k^2})$ convergence rate established in Theorem 14. In addition, the simulation also suggests that T is in general 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. Figures reporting test accuracy, and additional tests are postponed into Appendix.



(a1)

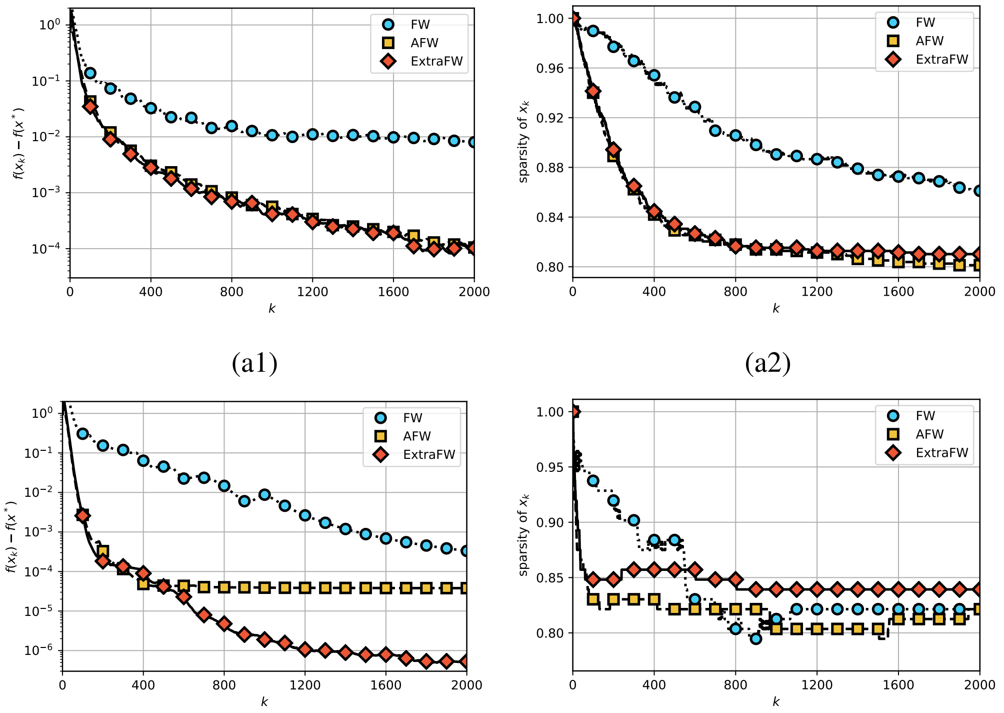
(a2)

(b1)

(b2)

Figure 4.2: 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*.

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 Appendix 4.5.11. 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 Figure 4.2. 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, with ExtraFW slightly faster than AFW. And ExtraFW finds sparser solutions than NAG.



(a1)

(a2)

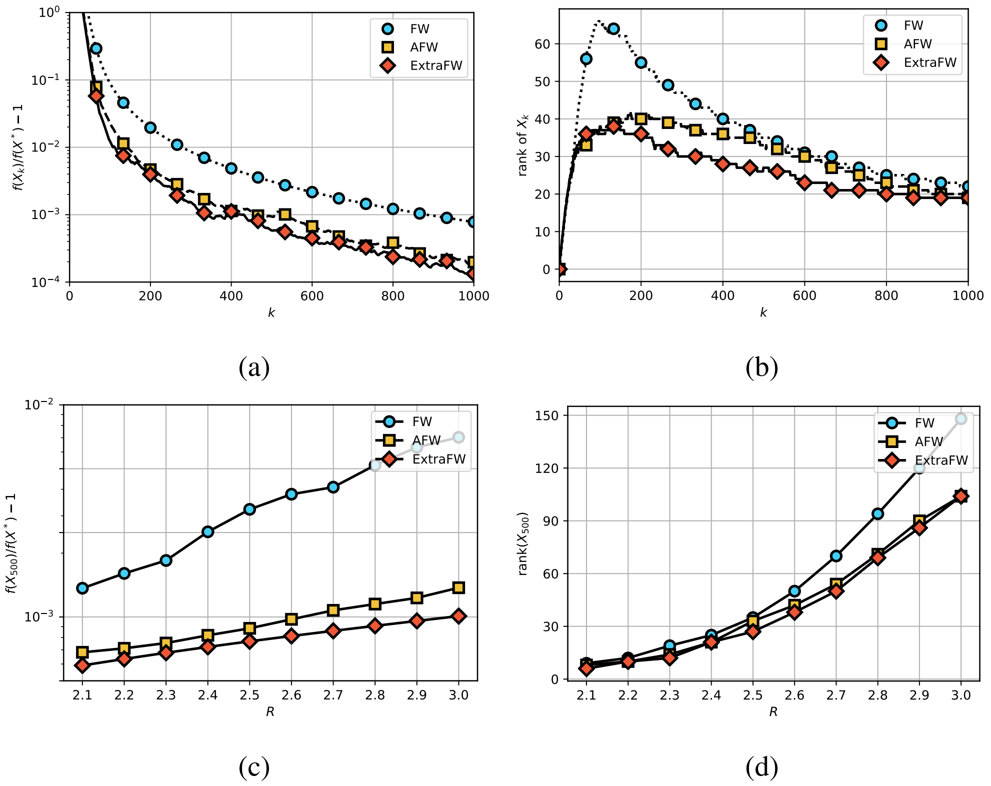
(b1)

(b2)

Figure 4.3: 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*.

n-support norm ball constraint. Effective projection onto such a constraint is unknown yet and hence GD and NAG are not included in the test. The performance of ExtraFW can be found in Figure 4.3. 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.

4.4.2 Matrix completion



(a)

(b)

(c)

(d)

Figure 4.4: 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.

We test ExtraFW on a widely used dataset, *MovieLens100K*[1] . The experiments follow the same steps in [18]. The numerical performance of ExtraFW, AFW, and FW can be found in Figure 4.4. We plot the optimality error and rank versus k choosing $R = 2.5$ in Figures 4.4(a) and 4.4(b). 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 4.4(c) and 4.4(d). 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

[1] `https://grouplens.org/datasets/movielens/100k/`

tendency of preferring ExtraFW over AFW and FW as R grows.

4.5 Appendix

4.5.1 Proof of Lemma 14

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 5. 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. □

4.5.2 Proof of Lemma 15

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

$$\begin{aligned}
\hat{\Phi}_{k+1}(\mathbf{x}) &= (1-\delta_k)\Phi_k(\mathbf{x}) + \delta_k \Big[f(\mathbf{y}_k) + \big\langle \nabla f(\mathbf{y}_k), \mathbf{x} - \mathbf{y}_k \big\rangle \Big] \\
&\leq (1-\delta_k)\Phi_k(\mathbf{x}) + \delta_k f(\mathbf{x}) \\
&\leq (1-\delta_k)\Big[(1-\lambda_k) f(\mathbf{x}) + \lambda_k \Phi_0(\mathbf{x}) \Big] + \delta_k f(\mathbf{x}) \\
&= (1-\lambda_{k+1}) f(\mathbf{x}) + \lambda_{k+1}\Phi_0(\mathbf{x}) \\
&= (1-\lambda_{k+1}) f(\mathbf{x}) + \lambda_{k+1}\hat{\Phi}_0(\mathbf{x}).
\end{aligned}$$

The proof is thus completed. □

4.5.3 Proof of Lemma 16

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

$$\begin{aligned}
\Phi_{k+1}(\mathbf{x}) &= (1-\delta_k)\Phi_k(\mathbf{x}) + \delta_k B_{k+1}(\mathbf{x}) \qquad (4.8) \\
&= (1-\delta_k)(1-\delta_{k-1})\Phi_{k-1}(\mathbf{x}) + (1-\delta_k)\delta_{k-1} B_k(\mathbf{x}) + \delta_k B_{k+1}(\mathbf{x}) \\
&= \Phi_0(\mathbf{x}) \prod_{\tau=0}^{k} (1-\delta_\tau) + \sum_{\tau=0}^{k} \delta_\tau B_{\tau+1}(\mathbf{x}) \prod_{j=\tau+1}^{k} (1-\delta_j) \\
&= f(\mathbf{x}_0) \prod_{\tau=0}^{k} (1-\delta_\tau) + \sum_{\tau=0}^{k} \delta_\tau B_{\tau+1}(\mathbf{x}) \prod_{j=\tau+1}^{k} (1-\delta_j).
\end{aligned}$$

Hence, the minimizer of $\Phi_{k+1}(\mathbf{x})$ can be rewritten as

$$\begin{aligned}
\arg\min_{\mathbf{x}\in\mathcal{X}} \ \Phi_{k+1}(\mathbf{x}) &= \arg\min_{\mathbf{x}\in\mathcal{X}} \ f(\mathbf{x}_0) \prod_{\tau=0}^{k} (1-\delta_\tau) + \sum_{\tau=0}^{k} \delta_\tau B_{\tau+1}(\mathbf{x}) \prod_{j=\tau+1}^{k} (1-\delta_j) \qquad (4.9) \\
&= \arg\min_{\mathbf{x}\in\mathcal{X}} \ \sum_{\tau=0}^{k} \delta_\tau \big[f(\mathbf{x}_{\tau+1}) + \langle \nabla f(\mathbf{x}_{\tau+1}), \mathbf{x} - \mathbf{x}_{\tau+1} \rangle \big] \prod_{j=\tau+1}^{k} (1-\delta_j) \\
&= \arg\min_{\mathbf{x}\in\mathcal{X}} \ \sum_{\tau=0}^{k} \delta_\tau \langle \nabla f(\mathbf{x}_{\tau+1}), \mathbf{x} \rangle \prod_{j=\tau+1}^{k} (1-\delta_j) \\
&= \arg\min_{\mathbf{x}\in\mathcal{X}} \ \sum_{\tau=0}^{k} \Big\langle \delta_\tau \nabla f(\mathbf{x}_{\tau+1}) \prod_{j=\tau+1}^{k} (1-\delta_j), \mathbf{x} \Big\rangle \\
&= \arg\min_{\mathbf{x}\in\mathcal{X}} \ \langle \mathbf{g}_{k+1}, \mathbf{x} \rangle
\end{aligned}$$

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 (4.9) 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

$$\begin{aligned}\hat{\Phi}_{k+1}(\mathbf{x}) &= (1-\delta_k)\Phi_k(\mathbf{x}) + \delta_k\Big[f(\mathbf{y}_k) + \big\langle \nabla f(\mathbf{y}_k), \mathbf{x}-\mathbf{y}_k\big\rangle\Big]\\ &= f(\mathbf{x}_0)\prod_{\tau=0}^{k}(1-\delta_\tau) + \sum_{\tau=0}^{k-1}\delta_\tau B_{\tau+1}(\mathbf{x}) \prod_{j=\tau+1}^{k}(1-\delta_j) + \delta_k\Big[f(\mathbf{y}_k) + \big\langle \nabla f(\mathbf{y}_k), \mathbf{x}-\mathbf{y}_k\big\rangle\Big].\end{aligned}$$

Hence using similar arguments as above we have

$$\begin{aligned}\arg\min_{\mathbf{x}\in\mathcal{X}}\ \hat{\Phi}_{k+1}(\mathbf{x}) &= \arg\min_{\mathbf{x}\in\mathcal{X}}\ \Big\langle \delta_k\nabla f(\mathbf{y}_k) + \sum_{\tau=0}^{k-1}\delta_\tau \nabla f(\mathbf{x}_{\tau+1}) \prod_{j=\tau+1}^{k}(1-\delta_j), \mathbf{x}\Big\rangle\\ &= \arg\min_{\mathbf{x}\in\mathcal{X}}\ \langle \hat{\mathbf{g}}_{k+1}, \mathbf{x}\rangle = \hat{\mathbf{v}}_{k+1}\end{aligned}$$

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 proved. □

4.5.4 Proof of Lemma 17

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 $f(\mathbf{x}_0) \leq \Phi_0^* + \xi_0$.

Now suppose that $f(\mathbf{x}_k) \leq \Phi_k^* + \xi_k$ 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 1 that

$$\begin{aligned}f(\mathbf{x}_{k+1}) &\leq f(\mathbf{y}_k) + \big\langle \nabla f(\mathbf{y}_k), \mathbf{x}_{k+1}-\mathbf{y}_k\big\rangle + \frac{L}{2}\|\mathbf{x}_{k+1}-\mathbf{y}_k\|^2 \qquad (4.10)\\ &\overset{(a)}{=} f(\mathbf{y}_k) + (1-\delta_k)\big\langle \nabla f(\mathbf{y}_k), \mathbf{x}_k-\mathbf{y}_k\big\rangle + \delta_k\big\langle \nabla f(\mathbf{y}_k), \hat{\mathbf{v}}_{k+1}-\mathbf{y}_k\big\rangle + \frac{L}{2}\|\mathbf{x}_{k+1}-\mathbf{y}_k\|^2\\ &\overset{(b)}{=} f(\mathbf{y}_k) + (1-\delta_k)\big\langle \nabla f(\mathbf{y}_k), \mathbf{x}_k-\mathbf{y}_k\big\rangle + \delta_k\big\langle \nabla f(\mathbf{y}_k), \hat{\mathbf{v}}_{k+1}-\mathbf{y}_k\big\rangle + \frac{L\delta_k^2}{2}\|\hat{\mathbf{v}}_{k+1}-\mathbf{v}_k\|^2\\ &\overset{(c)}{\leq} (1-\delta_k)f(\mathbf{x}_k) + \delta_k f(\mathbf{y}_k) + \delta_k\big\langle \nabla f(\mathbf{y}_k), \hat{\mathbf{v}}_{k+1}-\mathbf{y}_k\big\rangle + \frac{L\delta_k^2}{2}\|\hat{\mathbf{v}}_{k+1}-\mathbf{v}_k\|^2\end{aligned}$$

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 16). 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 (4.10) in; and the last inequality is because of Assumption 3. Rearrange the terms, we have

$$\begin{aligned}
f(\mathbf{x}_{k+1}) &\leq \hat{\Phi}_{k+1}^* + \frac{LD^2\delta_k^2}{2} + (1-\delta_k)\xi_k \\
&= \Phi_{k+1}^* + (\hat{\Phi}_{k+1}^* - \Phi_{k+1}^*) + \frac{LD^2\delta_k^2}{2} + (1-\delta_k)\xi_k.
\end{aligned} \tag{4.11}$$

Then, we have from Lemma 16 that

$$\begin{aligned}
\hat{\Phi}_{k+1}^* - \Phi_{k+1}^* &= s\hat{\Phi}_{k+1}(\hat{\mathbf{v}}_{k+1}) - \Phi_{k+1}(\mathbf{v}_{k+1}) \\
&= \hat{\Phi}_{k+1}(\hat{\mathbf{v}}_{k+1}) - \hat{\Phi}_{k+1}(\mathbf{v}_{k+1}) + \hat{\Phi}_{k+1}(\mathbf{v}_{k+1}) - \Phi_{k+1}(\mathbf{v}_{k+1}) \\
&\overset{(h)}{\leq} \hat{\Phi}_{k+1}(\mathbf{v}_{k+1}) - \Phi_{k+1}(\mathbf{v}_{k+1}) \\
&\overset{(i)}{=} \delta_k \Big[f(\mathbf{y}_k) + \big\langle \nabla f(\mathbf{y}_k), \mathbf{v}_{k+1} - \mathbf{y}_k \big\rangle \Big] - \delta_k \Big[f(\mathbf{x}_{k+1}) + \big\langle \nabla f(\mathbf{x}_{k+1}), \mathbf{v}_{k+1} - \mathbf{x}_{k+1} \big\rangle \Big] \\
&\overset{(j)}{\leq} \delta_k \big\langle \nabla f(\mathbf{y}_k) - \nabla f(\mathbf{x}_{k+1}), \mathbf{v}_{k+1} - \mathbf{x}_{k+1} \big\rangle \\
&\leq \delta_k \big\| \nabla f(\mathbf{y}_k) - \nabla f(\mathbf{x}_{k+1}) \big\|_* \big\| \mathbf{v}_{k+1} - \mathbf{x}_{k+1} \big\| \\
&\overset{(k)}{\leq} \delta_k L \big\| \mathbf{y}_k - \mathbf{x}_{k+1} \big\| \big\| \mathbf{v}_{k+1} - \mathbf{x}_{k+1} \big\| \\
&\overset{(l)}{\leq} \delta_k^2 L \big\| \mathbf{v}_k - \hat{\mathbf{v}}_{k+1} \big\| \big\| \mathbf{v}_{k+1} - \mathbf{x}_{k+1} \big\| \leq \delta_k^2 L D^2
\end{aligned} \tag{4.12}$$

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 16; (i) follows from (4.4); (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 1; and (l) uses the choice of $\mathbf{y}_k$ and $\mathbf{x}_{k+1}$. Plugging (4.12) back into (4.11), 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. □

4.5.5 Proof of Theorem 13

Proof. Given $\big(\{\Phi_k(\mathbf{x})\}_{k=0}^{\infty}, \{\lambda_k\}_{k=0}^{\infty}\big)$ is an ES as shown in Lemma 15, 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 17, one can directly apply Lemma 14 to have

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \lambda_k\big(f(\mathbf{x}_0) - f(\mathbf{x}^*)\big) + \xi_k = \frac{2\big(f(\mathbf{x}_0) - f(\mathbf{x}^*)\big)}{(k+1)(k+2)} + \xi_k \tag{4.13}$$

where ξ_k is defined in Lemma 17. 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 (4.13) completes the proof. □

4.5.6 Stopping criterion

In this subsection we show that the value of $f(\mathbf{x}_k) - \Phi_k^*$ can be used to derive a stopping criterion (see (4.14)). How to obtain the value of Φ_k^* iteratively (via (4.15) and (4.16)) is also discussed.

First, as a consequence of Lemma 17, 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

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{1}{1-\lambda_k}\bigg(f(\mathbf{x}_k) - \Phi_k^* - \lambda_k\big[f(\mathbf{x}_k) - f(\mathbf{x}_0)\big]\bigg). \tag{4.14}$$

Notice that the RHS of (4.14) goes to 0 as k increases, hence (4.14) can be used as the stopping criterion.

Finally we discuss how to update Φ_k^* efficiently. From (4.8), we have

$$\begin{aligned}\Phi_{k+1}(\mathbf{x}) &= f(\mathbf{x}_0)\prod_{\tau=0}^{k}(1-\delta_\tau) + \sum_{\tau=0}^{k}\delta_\tau\Big[f(\mathbf{x}_{\tau+1}) + \langle\nabla f(\mathbf{x}_{\tau+1}), \mathbf{x} - \mathbf{x}_{\tau+1}\rangle\Big]\prod_{j=\tau+1}^{k}(1-\delta_j)\\ &= f(\mathbf{x}_0)\prod_{\tau=0}^{k}(1-\delta_\tau) + \sum_{\tau=0}^{k}\delta_\tau\Big[f(\mathbf{x}_{\tau+1}) + \langle\nabla f(\mathbf{x}_{\tau+1}), \mathbf{x} - \mathbf{x}_{\tau+1}\rangle\Big]\prod_{j=\tau+1}^{k}(1-\delta_j)\\ &= f(\mathbf{x}_0)\prod_{\tau=0}^{k}(1-\delta_\tau) + \sum_{\tau=0}^{k}\delta_\tau\Big[f(\mathbf{x}_{\tau+1}) - \langle\nabla f(\mathbf{x}_{\tau+1}), \mathbf{x}_{\tau+1}\rangle\Big]\prod_{j=\tau+1}^{k}(1-\delta_j) + \langle\mathbf{g}_{k+1}, \mathbf{x}\rangle\end{aligned}$$

where the last equation uses the definition of $\mathbf{g}_{k+1}$. Hence, we can obtain Φ_{k+1}^* as

$$\Phi_{k+1}^* = \Phi_{k+1}(\mathbf{v}_{k+1}) = V_{k+1} + \langle\mathbf{g}_{k+1}, \mathbf{v}_{k+1}\rangle \tag{4.15}$$

and V_{k+1} can be updated as

$$V_{k+1} = (1-\delta_k)V_k + \delta_k\Big[f(\mathbf{x}_{k+1}) - \langle\nabla f(\mathbf{x}_{k+1}), \mathbf{x}_{k+1}\rangle\Big], \text{ with } V_0 = f(\mathbf{x}_0). \tag{4.16}$$

4.5.7 Proof of Theorem 14

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., $f : \tilde{\mathcal{X}} \rightarrow \mathbb{R}$. This is a very mild assumption since most of practically used loss functions have domain $\mathbb{R}^d$.

Lemma 18. *[8, Theorem 2.1.5] If Assumptions 1 and 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 19. *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 18, we have

$$\frac{1}{2L}\|\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}^*)\|_2^2 \leq f(\mathbf{x}_k) - f(\mathbf{x}^*) - \langle \nabla f(\mathbf{x}^*), \mathbf{x}_k - \mathbf{x}^* \rangle \overset{(a)}{\leq} f(\mathbf{x}_k) - f(\mathbf{x}^*)$$
$$\overset{(b)}{\leq} \frac{2\big(f(\mathbf{x}_0) - f(\mathbf{x}^*)\big)}{(k+1)(k+2)} + \frac{6LD^2}{k+2}$$

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 13. 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 20. *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 18, we have

$$\frac{1}{2L}\|\nabla f(\mathbf{x}_2^*) - \nabla f(\mathbf{x}_1^*)\|_2^2 \leq f(\mathbf{x}_2^*) - f(\mathbf{x}_1^*) - \langle \nabla f(\mathbf{x}_1^*), \mathbf{x}_2^* - \mathbf{x}_1^* \rangle \overset{(a)}{\leq} f(\mathbf{x}_2^*) - f(\mathbf{x}_1^*) = 0$$

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 21. *Let $\|\nabla f(\mathbf{x}^*)\|_2 = G^*$, (and G^* is unique bacause of Lemma 20) 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

$$\mathbf{g}_{k+1} = (1-\delta_k)\mathbf{g}_k + \delta_k \nabla f(\mathbf{x}_{k+1}) = \sum_{\tau=0}^{k} \delta_\tau \nabla f(\mathbf{x}_{\tau+1}) \Big[\prod_{j=\tau+1}^{k} (1-\delta_j) \Big] \tag{4.17}$$
$$= \sum_{\tau=0}^{k} \frac{2(\tau+2)}{(k+2)(k+3)} \nabla f(\mathbf{x}_{\tau+1}).$$

Noticing that $2\sum_{\tau=0}^{k}(\tau+2) = (k+1)(k+4) = (k+2)(k+3) - 2$, we have

$$\begin{aligned}
\|\mathbf{g}_{k+1} - \nabla f(\mathbf{x}^*)\|_2 &= \Big\| \sum_{\tau=0}^{k} \frac{2(\tau+2)}{(k+2)(k+3)} \big[\nabla f(\mathbf{x}_{\tau+1}) - \nabla f(\mathbf{x}^*)\big] - \frac{2}{(k+2)(k+3)} \nabla f(\mathbf{x}^*) \Big\|_2 \\
&\leq \sum_{\tau=0}^{k} \frac{2(\tau+2)}{(k+2)(k+3)} \big\|\nabla f(\mathbf{x}_{\tau+1}) - \nabla f(\mathbf{x}^*)\big\|_2 + \frac{2}{(k+2)(k+3)} \big\|\nabla f(\mathbf{x}^*)\big\|_2 \\
&\overset{(a)}{\leq} \sum_{\tau=0}^{k} \frac{2(\tau+2)}{(k+2)(k+3)} \frac{C_1}{\sqrt{\tau+3}} + \frac{2G^*}{(k+2)(k+3)} \\
&\leq \frac{2C_1}{(k+2)(k+3)} \sum_{\tau=0}^{k} \sqrt{\tau+2} + \frac{2G^*}{(k+2)(k+3)} \\
&\leq \frac{4C_1}{3(k+2)(k+3)} (k+3)^{3/2} + \frac{2G^*}{(k+2)(k+3)} \\
&= \frac{4C_1}{3(\sqrt{k+3}+1)(\sqrt{k+3}-1)} \sqrt{k+3} + \frac{2G^*}{(k+2)(k+3)} \\
&\leq \frac{4C_1}{3(\sqrt{k+3}-1)} + \frac{2G^*}{(k+2)(k+3)}
\end{aligned}$$

where (a) follows from Lemma 19. This completes the proof for the first part of this lemma. Next, to find C_2, we have

$$\begin{aligned}
\|\mathbf{g}_{k+1} - \nabla f(\mathbf{x}^*)\|_2 &\leq \frac{4C_1}{3(\sqrt{k+3}-1)} + \frac{2G^*}{(k+2)(k+3)} \\
&= \frac{4C_1}{3(\sqrt{k+3}-1)} + \frac{2G^*}{(k+3)(\sqrt{k+3}+1)(\sqrt{k+3}-1)} \\
&\overset{(b)}{\leq} \frac{4C_1}{3(\sqrt{k+3}-1)} + \frac{2G^*}{3(\sqrt{3}+1)(\sqrt{k+3}-1)}
\end{aligned}$$

where in (b) we use $k+3 \geq 3$ and $\sqrt{k+3}+1 \geq \sqrt{3}+1$. The proof is thus completed. □

Lemma 22. *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 21, 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 proved. □

Lemma 23. *Let* $T := \max\{T_1, T_2\}$, *with* $T_2 = \sqrt{\frac{8LD}{G^*}} - 3$. *When* $k \geq T + 1$, *it is guaranteed that*

$$\|\mathbf{v}_{k+1} - \hat{\mathbf{v}}_{k+1}\|_2 \leq \frac{\delta_k^3 LDC_3}{\|\mathbf{g}_{k+1}\|_2 \|\mathbf{g}_k\|_2} \leq \frac{4\delta_k^3 LDC_3}{(G^*)^2} \tag{4.18}$$

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 22 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 (4.5), we can write

$$\begin{aligned}
&\|\mathbf{v}_{k+1} - \hat{\mathbf{v}}_{k+1}\|_2 = \left\| -\frac{R}{\|\mathbf{g}_{k+1}\|_2}\mathbf{g}_{k+1} + \frac{R}{\|\hat{\mathbf{g}}_{k+1}\|_2}\hat{\mathbf{g}}_{k+1} \right\|_2 \qquad (4.19) \\
&= \frac{R}{\|\mathbf{g}_{k+1}\|_2 \|\hat{\mathbf{g}}_{k+1}\|_2} \left\| \big\|\hat{\mathbf{g}}_{k+1}\big\|_2 \mathbf{g}_{k+1} - \big\|\mathbf{g}_{k+1}\big\|_2 \hat{\mathbf{g}}_{k+1} \right\|_2 \\
&= \frac{R}{\|\mathbf{g}_{k+1}\|_2 \|\hat{\mathbf{g}}_{k+1}\|_2} \left\| \big\|\hat{\mathbf{g}}_{k+1}\big\|_2 \mathbf{g}_{k+1} - \big\|\hat{\mathbf{g}}_{k+1}\big\|_2 \hat{\mathbf{g}}_{k+1} + \big\|\hat{\mathbf{g}}_{k+1}\big\|_2 \hat{\mathbf{g}}_{k+1} - \big\|\mathbf{g}_{k+1}\big\|_2 \hat{\mathbf{g}}_{k+1} \right\|_2 \\
&\leq \frac{R}{\|\mathbf{g}_{k+1}\|_2} \left\| \mathbf{g}_{k+1} - \hat{\mathbf{g}}_{k+1} \right\|_2 + \frac{R}{\|\mathbf{g}_{k+1}\|_2} \left| \big\|\hat{\mathbf{g}}_{k+1}\big\|_2 - \big\|\mathbf{g}_{k+1}\big\|_2 \right| \\
&\overset{(a)}{\leq} \frac{2R}{\|\mathbf{g}_{k+1}\|_2} \left\| \mathbf{g}_{k+1} - \hat{\mathbf{g}}_{k+1} \right\|_2 = \frac{2R\delta_k}{\|\mathbf{g}_{k+1}\|_2} \left\| \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{y}_k) \right\|_2 \\
&\overset{(b)}{\leq} \frac{2RL\delta_k}{\|\mathbf{g}_{k+1}\|_2} \left\| \mathbf{x}_{k+1} - \mathbf{y}_k \right\|_2 = \frac{DL\delta_k^2}{\|\mathbf{g}_{k+1}\|_2} \left\| \hat{\mathbf{v}}_{k+1} - \mathbf{v}_k \right\|_2
\end{aligned}$$

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 1. Then we will bound $\|\hat{\mathbf{v}}_{k+1} - \mathbf{v}_k\|_2$.

$$\begin{aligned}
&\big\|\hat{\mathbf{v}}_{k+1} - \mathbf{v}_k\big\|_2 = \bigg\| - \frac{R}{\|\hat{\mathbf{g}}_{k+1}\|_2}\hat{\mathbf{g}}_{k+1} + \frac{R}{\|\mathbf{g}_k\|_2}\mathbf{g}_k\bigg\|_2 \\
&= \frac{R}{\|\mathbf{g}_k\|_2\|\hat{\mathbf{g}}_{k+1}\|_2}\bigg\|\big\|\mathbf{g}_k\big\|_2\hat{\mathbf{g}}_{k+1} - \big\|\hat{\mathbf{g}}_{k+1}\big\|_2\hat{\mathbf{g}}_{k+1} + \big\|\hat{\mathbf{g}}_{k+1}\big\|_2\hat{\mathbf{g}}_{k+1} - \big\|\hat{\mathbf{g}}_{k+1}\big\|_2\mathbf{g}_k\bigg\|_2 \\
&\leq \frac{R}{\|\mathbf{g}_k\|_2}\bigg|\big\|\mathbf{g}_k\big\|_2 - \big\|\hat{\mathbf{g}}_{k+1}\big\|_2\bigg| + \frac{R}{\|\mathbf{g}_k\|_2}\bigg\|\hat{\mathbf{g}}_{k+1} - \mathbf{g}_k\bigg\|_2 \\
&\overset{(c)}{\leq} \frac{D}{\|\mathbf{g}_k\|_2}\bigg\|\hat{\mathbf{g}}_{k+1} - \mathbf{g}_k\bigg\|_2 = \frac{\delta_k D}{\|\mathbf{g}_k\|_2}\bigg\|\nabla f(\mathbf{y}_k) - \mathbf{g}_k\bigg\|_2 \\
&\leq \frac{\delta_k D}{\|\mathbf{g}_k\|_2}\big\|\nabla f(\mathbf{y}_k) - \nabla f(\mathbf{x}^*)\big\|_2 + \frac{\delta_k D}{\|\mathbf{g}_k\|_2}\big\|\nabla f(\mathbf{x}^*) - \mathbf{g}_k\big\|_2 \\
&\leq \frac{\delta_k LD^2}{\|\mathbf{g}_k\|_2} + \frac{\delta_k D}{\|\mathbf{g}_k\|_2}\big\|\nabla f(\mathbf{x}^*) - \mathbf{g}_k\big\|_2 \\
&\leq \frac{\delta_k LD^2}{\|\mathbf{g}_k\|_2} + \frac{\delta_k D}{\|\mathbf{g}_k\|_2}\frac{C_2}{\sqrt{k+2}-1} \leq \frac{\delta_k\big(LD^2 + \frac{DC_2}{\sqrt{T+3}-1}\big)}{\|\mathbf{g}_k\|_2} := \frac{\delta_k C_3}{\|\mathbf{g}_k\|_2}
\end{aligned}$$

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 19. Plugging back to (4.19), 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 24. *Let $\xi_0 = 0$ and T defined the same as in Lemma 23. 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 17, hence it is omitted here. We mainly focus on the case where $k \geq T+1$.

$$
\begin{aligned}
\Phi_{k+1}^* = \Phi_{k+1}(\mathbf{v}_{k+1}) &= (1-\delta_k)\Phi_k(\mathbf{v}_{k+1}) + \delta_k\Big[f(\mathbf{x}_{k+1}) + \big\langle \nabla f(\mathbf{x}_{k+1}), \mathbf{v}_{k+1} - \mathbf{x}_{k+1}\big\rangle\Big] \\
&\overset{(a)}{\geq} (1-\delta_k)\Phi_k(\mathbf{v}_k) + \delta_k\Big[f(\mathbf{x}_{k+1}) + \big\langle \nabla f(\mathbf{x}_{k+1}), \mathbf{v}_{k+1} - \mathbf{x}_{k+1}\big\rangle\Big] \\
&\geq (1-\delta_k)f(\mathbf{x}_k) + \delta_k\Big[f(\mathbf{x}_{k+1}) + \big\langle \nabla f(\mathbf{x}_{k+1}), \mathbf{v}_{k+1} - \mathbf{x}_{k+1}\big\rangle\Big] - (1-\delta_k)\xi_k \\
&= f(\mathbf{x}_{k+1}) + (1-\delta_k)\big[f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})\big] + \delta_k\big\langle \nabla f(\mathbf{x}_{k+1}), \mathbf{v}_{k+1} - \mathbf{x}_{k+1}\big\rangle - (1-\delta_k)\xi_k \\
&\overset{(b)}{\geq} f(\mathbf{x}_{k+1}) + (1-\delta_k)\big\langle \nabla f(\mathbf{x}_{k+1}), \mathbf{x}_k - \mathbf{x}_{k+1}\big\rangle + \delta_k\big\langle \nabla f(\mathbf{x}_{k+1}), \mathbf{v}_{k+1} - \mathbf{x}_{k+1}\big\rangle - (1-\delta_k)\xi_k \\
&= f(\mathbf{x}_{k+1}) + \delta_k\big\langle \nabla f(\mathbf{x}_{k+1}), \mathbf{v}_{k+1} - \hat{\mathbf{v}}_{k+1}\big\rangle - (1-\delta_k)\xi_k \\
&\overset{(c)}{\geq} f(\mathbf{x}_{k+1}) - \delta_k\|\nabla f(\mathbf{x}_{k+1})\|_2\|\mathbf{v}_{k+1} - \hat{\mathbf{v}}_{k+1}\|_2 - (1-\delta_k)\xi_k \\
&\overset{(d)}{\geq} f(\mathbf{x}_{k+1}) - \|\nabla f(\mathbf{x}_{k+1})\|_2\frac{4\delta_k^4 LDC_3}{(G^*)^2} - (1-\delta_k)\xi_k \\
&\overset{(e)}{\geq} f(\mathbf{x}_{k+1}) - \Big(\frac{C_1}{\sqrt{T+4}} + G^*\Big)\frac{4\delta_k^4 LDC_3}{(G^*)^2} - (1-\delta_k)\xi_k
\end{aligned}
$$

where (a) is because $\mathbf{v}_k$ minimizes $\Phi_k(\mathbf{x})$ shown in Lemma 16; (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 23, 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 19. □

Proof of Theorem 14

Proof. Let T be defined the same as in Lemma 22. 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

$$\begin{aligned}
\xi_{k+1} &= (1-\delta_k)\xi_k + \theta_k = \sum_{\tau=0}^{k} \theta_\tau \prod_{j=\tau+1}^{k} (1-\delta_j) \\
&= \sum_{\tau=0}^{k} \theta_\tau \frac{(\tau+2)(\tau+3)}{(k+2)(k+3)} \\
&= \sum_{\tau=0}^{T} \frac{3LD^2}{2} \delta_\tau^2 \frac{(\tau+2)(\tau+3)}{(k+2)(k+3)} + \sum_{\tau=T+1}^{k} C_4 \delta_\tau^4 \frac{(\tau+2)(\tau+3)}{(k+2)(k+3)} \\
&= \frac{6LD^2(T+1)}{(k+2)(k+3)} + \mathcal{O}\bigg(\frac{C_4}{k^3}\bigg). \qquad (4.20)
\end{aligned}$$

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 14, we have

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{2\big[f(\mathbf{x}_0) - f(\mathbf{x}^*)\big]}{(k+1)(k+2)} + \xi_k. \qquad (4.21)$$

Plugging the expression of ξ_k, i.e., (4.20), into (4.21) completes the proof. □

4.5.8 ℓ_1 norm ball

In this subsection 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* [86], and it naturally implies that the constraint is active). Note that in this case Lemma 20 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 (4.6). The constants required in the proof is summarized below for clearance. The norm considered in this subsection 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 25. *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 21 still holds.

We first show that there exist $T_1 = (\frac{3C_2}{\lambda} + 1)^2 - 3$, 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 21 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 21, we have

$$\big\|\hat{\mathbf{g}}_{k+1} - \nabla f(\mathbf{x}^*)\big\|_2 = \big\|(1-\delta_k)\mathbf{g}_k + \delta_k \nabla f(\mathbf{x}_{k+1}) - \delta_k \nabla f(\mathbf{x}_{k+1}) + \delta_k \nabla f(\mathbf{y}_k) - \nabla f(\mathbf{x}^*)\big\|_2$$
$$\leq \big\|\mathbf{g}_{k+1} - \nabla f(\mathbf{x}^*)\|_2 + \delta_k \|\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{y}_k)\big\|_2 \leq \big\|\mathbf{g}_{k+1} - \nabla f(\mathbf{x}^*)\|_2 + \delta_k^2 L_2 D_2$$
$$\leq \frac{C_2}{\sqrt{k+3}-1} + \frac{4L_2D_2}{(k+3)^2} \leq \frac{C_5}{\sqrt{k+3}-1}, \forall k \geq T_1$$

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 24 to cope with the ℓ_1 norm ball constraint.

Lemma 26. *Let $\xi_0 = 0$ and T be the same as in Lemma 25. 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 17, hence it is omitted here. We mainly focus on the case where $k \geq T$. Using similar argument as in Lemma 24, 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 25. □

Theorem 15. *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 1 - 3 are satisfied, ExtraFW guarantees that*

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) = \mathcal{O}\Big(\frac{1}{k^2}\Big).$$

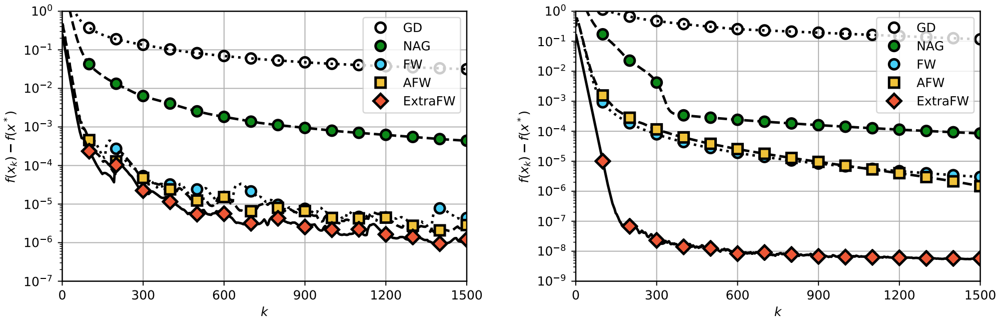
Proof. Let T be defined the same as in Lemma 25. For convenience denote $\xi_{k+1} = (1-\delta_k)\xi_k + \theta_k$. When $k < T$, we have $\theta_k = \frac{3LD^2}{2}\delta_k^2$; when $k \geq T$, we have $\theta_k = 0$. Then we can write

$$\begin{aligned}\xi_{k+1} = (1-\delta_k)\xi_k + \theta_k &= \sum_{\tau=0}^{k} \theta_\tau \prod_{j=\tau+1}^{k} (1-\delta_j) = \sum_{\tau=0}^{k} \theta_\tau \frac{(\tau+2)(\tau+3)}{(k+2)(k+3)} \\ &= \sum_{\tau=0}^{T-1} \frac{3LD^2}{2}\delta_\tau^2 \frac{(\tau+2)(\tau+3)}{(k+2)(k+3)} = \frac{6LD^2 T}{(k+2)(k+3)}.\end{aligned} \tag{4.22}$$

Finally, applying Lemma 14, we have

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{2\big[f(\mathbf{x}_0) - f(\mathbf{x}^*)\big]}{(k+1)(k+2)} + \xi_k. \tag{4.23}$$

Plugging the expression of ξ_k, i.e., (4.22) into (4.23) completes the proof. □



(a) *mnist*

(b) *mushroom*

Figure 4.5: ExtraFW guarantees an $\mathcal{O}(\frac{1}{k^2})$ rate on simplex.

Beyond ℓ_1 norm ball. The $\mathcal{O}(\frac{T}{k^2})$ rate in Theorem 15 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 26 to obtain the $\mathcal{O}(\frac{T}{k^2})$ rate. Numerical evidences using logistic regression as objective function can be found in Figure 4.5. Note that in this case however, FW itself converges fast enough.

4.5.9 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 14 and 15, 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 1 - 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 $G_n^* := \|\text{top}_n(\nabla f(\mathbf{x}^*))\|_2$ helpful. Clearly, $G^* \geq G_n^* \geq \sqrt{\frac{n}{d}}G^*$.

Theorem 16. *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 1 - 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 25, 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 22, one can show that there exists a constant $\tilde{T}_2$ such that $\|\text{top}_n(\mathbf{g}_{k+1})\|_2 \geq \frac{G_n^*}{2}$.

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 $\|\text{top}_n(\mathbf{g}_{k+1})\|_2 \geq \frac{G_n^*}{2}, \forall k \geq T_2$, 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 23.

Finally, using similar arguments of Lemma 24 with the aid of $\|\mathbf{v}_{k+1} - \hat{\mathbf{v}}_{k+1}\|_2 = \mathcal{O}(\delta_k^3)$, and applying Lemma 14, we can obtain $f(\mathbf{x}_k) - f(\mathbf{x}^*) = \mathcal{O}\big(\frac{T_2}{k^2}\big)$. □

4.5.10 Additional Numerical Results

All numerical experiments are performed using Python 3.7 on an Intel i7-4790CPU @3.60 GHz (32 GB RAM) desktop.

Efficiency of ExtraFW: Case Study of n-support Norm Ball In this subsection 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 4.3.3, that is

$$\min_{\mathbf{x}} \; f(\mathbf{x}) + \lambda(\|\mathbf{x}\|_{\mathrm{n-sp}})^2 \tag{4.24a}$$

$$\Leftrightarrow \quad \min_{\mathbf{x}} \; f(\mathbf{x}) \;\; \text{s.t.} \;\; \|\mathbf{x}\|_{\mathrm{n-sp}} \leq R \tag{4.24b}$$

where $\|\cdot\|_{\mathrm{n-sp}}$ denotes the n-support norm [12].

Clearly, one can apply proximal NAG (Prox-NAG) to (4.24a). The proximal operator per iteration has complexity $\mathcal{O}(d(n + \log d))$ [12].

One can also apply ExtraFW for (4.24b). From the Lagrangian duality of (4.24b) and (4.24a), one can see that if $\lambda \neq 0$, one must have an optimal solution for (4.24b) 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 4.1: A comparison of different algorithms for logistic regression with n-support norm

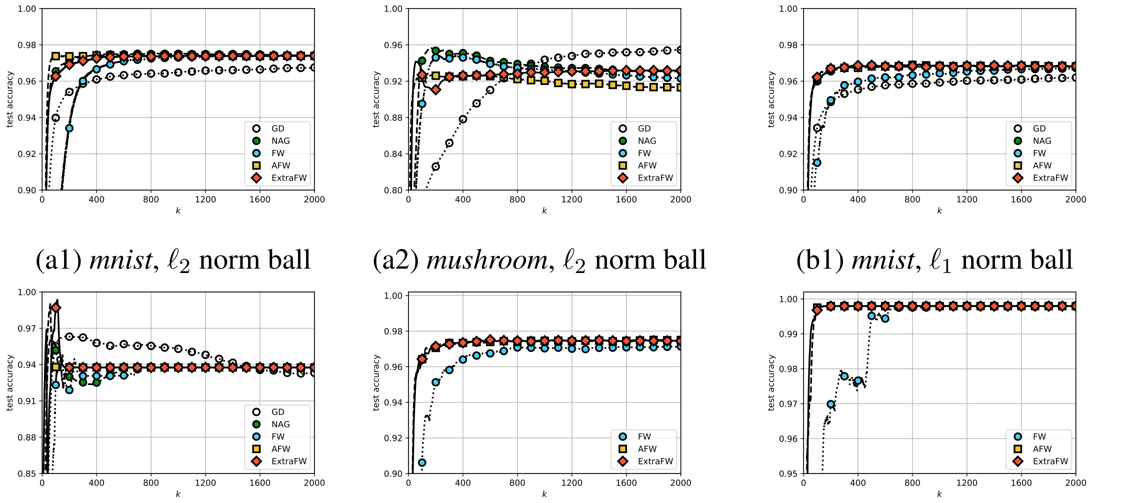
Alg.	convergence rate	per iteration cost
Prox-NAG for (4.24a)	$\mathcal{O}(1/k^2)$	proximal operator: $\mathcal{O}(d(n + \log d))$
Projected NAG for (4.24b)	$\mathcal{O}(1/k^2)$	projection is expensive
FW for (4.24b)	$\mathcal{O}(1/k)$	FW step: $\mathcal{O}(d \log n)$
ExtraFW for (4.24b)	$\mathcal{O}(T/k^2)$	FW step: $\mathcal{O}(d \log n)$

4.5.11 Binary classification

Table 4.2: A summary of datasets used in numerical tests

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 tests are summarized in Table 4.2.

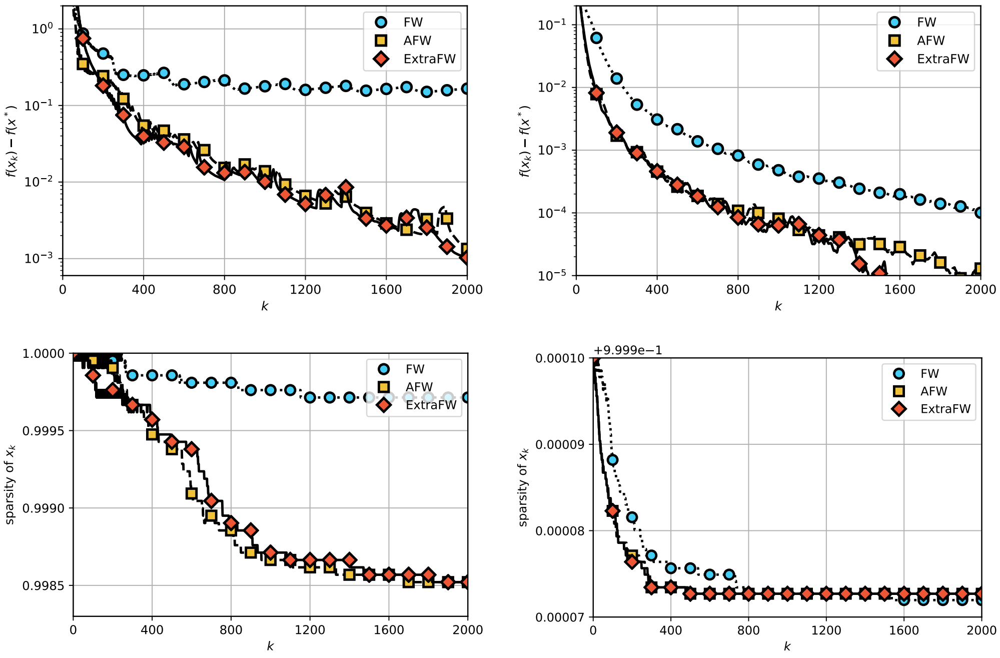


(a1) *mnist*, ℓ_2 norm ball (a2) *mushroom*, ℓ_2 norm ball (b1) *mnist*, ℓ_1 norm ball

(a2) *mushroom*, ℓ_1 norm ball (c1) *mnist*, n-supp norm ball (c2) *mushroom*, n-supp norm ball

Figure 4.6: Test accuracy of ExtraFW on different constraints.

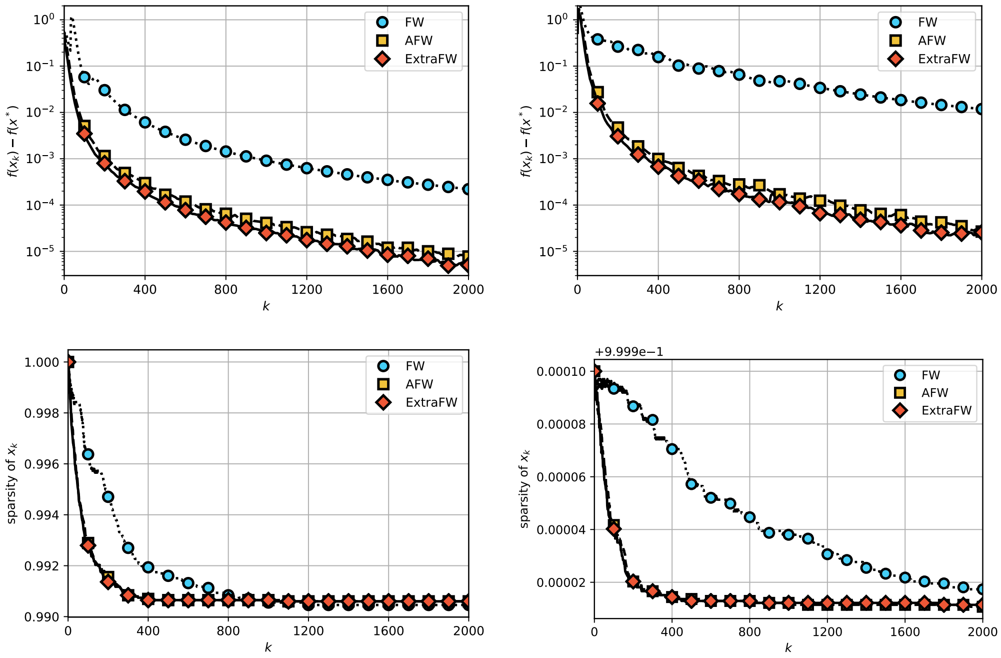
The test accuracy of different algorithms can be found in Figure 4.6. Additional numerical results for ℓ_1 norm ball constraint can be found in Figure 4.7. It can be seen that on dataset *realsim*, ExtraFW has similar performance with AFW, both outperforming FW significantly. On dataset *news20*, ExtraFW outperforms AFW in terms of optimality error.



(a) *realsim*

(b) *news20*

Figure 4.7: Additional tests of ExtraFW for classification with $\mathcal{X}$ being an ℓ_1 norm ball.



(a) *realsim*

(b) *news20*

Figure 4.8: Additional tests of ExtraFW for classification with $\mathcal{X}$ being an n-support norm ball.

Additional tests for n-support norm ball constraint are listed in Figure 4.8. The optimality error of ExtraFW is smaller than AFW on both *realsim* and *news20*.

Chapter 5

Summary and Future Directions

To conclude this dissertation, a summary of its main results and possible directions for future research are provided in this final chapter.

5.1 Thesis Summary

Chapter 2 demonstrated the merits of heavy ball momentum for FW (HFW). Multiple choices of the step size ensured a tighter Type II primal-dual error bound that can be efficiently computed when adopted as stopping criterion. An even tighter PD error bound can be achieved by relying jointly on heavy ball momentum and restart. A novel and general approach was developed to compute local Lipschitz constants in FW type algorithms.

In Chapter 3, we built links between Nesterov's momentum and the FW step by observing that they are both minimizing an (approximated) lower bound of the objective function. Exploring this link, we show how momentum benefits parameter-free FW. In particular, a momentum variant of FW, which we term AFW, was proved to achieve a faster rate on active ℓ_p norm ball constraints while maintaining the same convergence rate as FW on general problems. AFW thus strictly outperforms FW providing the possibility for acceleration.

A new parameter-free FW variant, ExtraFW, is introduced and analyzed in Chapter 4. ExtraFW leverages two gradient evaluations per iteration to update in a "prediction-correction" 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. The convergence rate of ExtraFW improves over AFW. Given the possibility of acceleration, ExtraFW is thus a competitive alternative to FW.

The efficiency of HFW, AFW, and ExtraFW is validated on tasks such as i) binary classification with different constraints, and ii) matrix completion problems. The numerical experiments aligns well with our theoretical findings, demonstrating the potential of momentum for FW type algorithms.

5.2 Future research

The results in this dissertation open up interesting directions for a number of future research topics, including both theories and applications. Next, we briefly discuss a couple of topics that we pursue currently.

- **Momentum for FW on polytopes.** While this dissertation copes with general convex constraint sets, it is known that there are FW variants with faster rates [44, 42], and those with leveraging oracle simpler than LMO [91, 92]. Therefore, one natural question to ask is that given a polytope constraint, can momentum help to make FW (variants) even more efficient?

- **Local smoothness in FW.** Estimate of the Lipschitz constant L is typically too pessimistic, this is the key reason that smooth step sizes do not perform well in our numerical tests. Although this problem can be partially solved via directional-smooth step sizes, it is not always easy to compute the directional-smooth parameter. We believe that it is possible to access the local smoothness in a more delicate and adaptive manner such as [93].

- **Momentum aided FW for stochastic optimization.** While this thesis demonstrates the benefit of momentum for batch methods, whether momentum is helpful for stochastic FW, e.g., [56, 57] is still unclear. It is intriguing to investigate and analyze related algorithms. As FW is sensitive to stochastic noise [7, 94, 95], understanding the performance of momentum in this setting is even more challenging. Another relating topic is to combine variance reduction techniques [96, 97, 98, 75, 99, 80, 100, 101] with momentum FW. As the stochastic noise is carefully controlled in such methods, intuitively it should be easier to use with stochastic FW with momentum.

- **Stochastic bandits from a FW point of view.** FW also links with other fundamental machine learning frameworks such as stochastic bandit [102]. Building upon this link, it might be beneficial to revisit classical bandit problems [103, 104, 105, 106] using a FW perspective. Hopefully, the fruitful results in FW literature can cross-fertilize bandit problems by providing more insights and alternative approaches to handle the classical problem.

- **Pruning for overparameterized deep neural networks (DNN).** The faster rates of AFW and ExtraFW in Chapters 3 and 4 will carry over to DNN pruning. Building upon [32], we hope to establish that AFW- and ExtraFW-based pruning will incur losses bounded by $\epsilon_n = \mathcal{O}(\min\{\frac{1}{n}, \frac{n_t}{n^2}\})$ for a certain problem-dependent constant n_t that has analytical form and n denoting the remaining neurons. As a result, the hope AFW and ExtraFW will accelerate DNN pruning. Another important implication of the envisioned bound on the loss is that ϵ_n is strictly smaller than $\mathcal{O}(\frac{1}{n})$ when $n \geq n_t$. This suggests that DNN pruning benefits from overparametrization, meaning that it is useful to set the number of neurons in an un-pruned network to satisfy $N > n \geq n_t$. This result corroborates a recent study [32], which also points out that overparametrization is helpful for pruning DNNs. While [32] does not quantify *how many neurons should comprise an overparametrized DNN*, our bound on ϵ_n establishes that at least n_t neurons are required.

- **Frank Wolfe for fairness constraints.** Recent years have witnessed the pressing need to develop algorithms that satisfy fair, responsible and trustworthy requirements on machine learning models. One of the solutions is to pose fair constraints [107]. We hope to develop FW variants that are tailored for such fair constraints. Our vision is that the FW subproblem can be explainable, making the obtained model more convincing and reasonable.

References

[1] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

[2] Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proc. Intl. Conf. on Machine Learning*, pages 427–435, 2013.

[3] Guanghui Lan. The complexity of large-scale convex programming under a linear optimization oracle. *arXiv preprint arXiv:1309.5550*, 2013.

[4] Robert M Freund and Paul Grigas. New analysis and results for the Frank–Wolfe method. *Mathematical Programming*, 155(1-2):199–230, 2016.

[5] Bingcong Li, Alireza Sadeghi, and Georgios Giannakis. Heavy ball momentum for conditional gradient. *Proc. Advances in Neural Info. Process. Syst.*, 34, 2021.

[6] Simon Lacoste-Julien. Convergence rate of frank-wolfe for non-convex objectives. *arXiv preprint arXiv:1607.00345*, 2016.

[7] Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Stochastic frank-wolfe methods for nonconvex optimization. In *2016 54th annual Allerton conference on communication, control, and computing (Allerton)*, pages 1244–1251. IEEE, 2016.

[8] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2004.

[9] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[10] Gabriel Peyré, Marco Cuturi, et al. Computational ooptimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[11] Xiangrong Zeng and Mário AT Figueiredo. Decreasing weighted sorted ℓ_1 regularization. *IEEE Signal Processing Letters*, 21(10):1240–1244, 2014.

[12] Andreas Argyriou, Rina Foygel, and Nathan Srebro. Sparse prediction with the k-support norm. In *Proc. Advances in Neural Info. Process. Syst.*, pages 1457–1465, 2012.

[13] Maryam Fazel. Matrix rank minimization with applications. 2002.

[14] James Bennett, Stan Lanning, et al. The Netflix prize. In *Proc. KDD cup and workshop*, volume 2007, page 35. New York, NY, USA., 2007.

[15] Robert M Bell and Yehuda Koren. Lessons from the Netflix prize challenge. *SiGKDD Explorations*, 9(2):75–79, 2007.

[16] Zeyuan Allen-Zhu, Elad Hazan, Wei Hu, and Yuanzhi Li. Linear convergence of a Frank-Wolfe type algorithm over trace-norm balls. In *Proc. Advances in Neural Info. Process. Syst.*, pages 6191–6200, 2017.

[17] Zaid Harchaoui, Anatoli Juditsky, and Arkadi Nemirovski. Conditional gradient algorithms for norm-regularized smooth convex optimization. *Mathematical Programming*, 152(1-2):75–112, 2015.

[18] Robert M Freund, Paul Grigas, and Rahul Mazumder. An extended Frank–Wolfe method with "in-face" directions, and its application to low-rank matrix completion. *SIAM Journal on Optimization*, 27(1):319–346, 2017.

[19] Liang Zhang, Vassilis Kekatos, and Georgios B Giannakis. A generalized frank-wolfe approach to decentralized electric vehicle charging. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 1105–1111. IEEE, 2016.

[20] Liang Zhang, Gang Wang, Daniel Romero, and Georgios B Giannakis. Randomized block Frank–Wolfe for convergent large-scale learning. *IEEE Transactions on Signal Processing*, 65(24):6448–6461, 2017.

[21] Yu-Xiang Wang, Veeranjaneyulu Sadhanala, Wei Dai, Willie Neiswanger, Suvrit Sra, and Eric Xing. Parallel and distributed block-coordinate frank-wolfe algorithms. In *Proc. Intl. Conf. on Machine Learning*, pages 1548–1557. PMLR, 2016.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Proc. Advances in Neural Info. Process. Syst.*, 30, 2017.

[24] Bingcong Li, Tianyi Chen, and Georgios B Giannakis. Secure mobile edge computing in iot via collaborative online learning. *IEEE Transactions on Signal Processing*, 67(23):5922–5935, 2019.

[25] Bingcong Li, Tianyi Chen, Xin Wang, and Georgios B Giannakis. Real-time energy management in microgrids with reduced battery capacity requirements. *IEEE Transactions on Smart Grid*, 10(2):1928–1938, 2017.

[26] Han Cai, Ligeng Zhu, and Song Han. ProxylessNAS: Direct neural architecture search on target task and hardware. *arXiv preprint arXiv:1812.00332*, 2018.

[27] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[28] Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural networks. *arXiv preprint arXiv:1506.02626*, 2015.

[29] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proc. IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 2736–2744, 2017.

[30] Ruichi Yu, Ang Li, Chun-Fu Chen, Jui-Hsin Lai, Vlad I Morariu, Xintong Han, Mingfei Gao, Ching-Yung Lin, and Larry S Davis. Nisp: Pruning networks using neuron importance score propagation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 9194–9203, 2018.

[31] Ben Mussay, Margarita Osadchy, Vladimir Braverman, Samson Zhou, and Dan Feldman. Data-independent neural pruning via coresets. *arXiv preprint arXiv:1907.04018*, 2019.

[32] Mao Ye, Chengyue Gong, Lizhen Nie, Denny Zhou, Adam Klivans, and Qiang Liu. Good subnetworks provably exist: Pruning via greedy forward selection. In *Proc. Intl. Conf. on Machine Learning*, 2020.

[33] Masao Fukushima. A modified Frank-Wolfe algorithm for solving the traffic assignment problem. *Transportation Research Part B: Methodological*, 18(2):169–177, 1984.

[34] Tri Nguyen, Xiao Fu, and Ruiyuan Wu. Memory-efficient convex optimization for self-dictionary separable nonnegative matrix factorization: A frank-wolfe approach. *arXiv preprint arXiv:2109.11135*, 2021.

[35] Armand Joulin, Kevin Tang, and Li Fei-Fei. Efficient image and video co-localization with Frank-Wolfe algorithm. In *Proc. European Conf. on Computer Vision*, pages 253–268. Springer, 2014.

[36] Simon Lacoste-Julien, Fredrik Lindsten, and Francis Bach. Sequential kernel herding: Frank-Wolfe optimization for particle filtering. In *Proc. Intl. Conf. on Artificial Intelligence and Statistics*, pages 544–552, 2015.

[37] Immanuel M Bomze, Francesco Rinaldi, and Damiano Zeffiro. Fast cluster detection in networks by first-order optimization. *arXiv preprint arXiv:2103.15907*, 2021.

[38] Jinghui Chen, Dongruo Zhou, Jinfeng Yi, and Quanquan Gu. A frank-wolfe framework for efficient and effective adversarial attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3486–3494, 2020.

[39] Leonard Berrada, Andrew Zisserman, and M Pawan Kumar. Deep frank-wolfe for neural network optimization. *arXiv preprint arXiv:1811.07591*, 2018.

[40] Giulia Luise, Saverio Salzo, Massimiliano Pontil, and Carlo Ciliberto. Sinkhorn barycenters with free support via Frank-Wolfe algorithm. In *Proc. Advances in Neural Info. Process. Syst.*, pages 9318–9329, 2019.

[41] Cyrille W Combettes and Sebastian Pokutta. Complexity of linear minimization and projection on some sets. *Operations Research Letters*, 49(4):565–571, 2021.

[42] Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of Frank-Wolfe optimization variants. In *Proc. Advances in Neural Info. Process. Syst.*, pages 496–504, 2015.

[43] Jacques Guélat and Patrice Marcotte. Some comments on Wolfe's 'away step'. *Mathematical Programming*, 35(1):110–119, 1986.

[44] BF Mitchell, Vladimir Fedorovich Dem'yanov, and VN Malozemov. Finding the point of a polyhedron closest to the origin. *SIAM Journal on Control*, 12(1):19–26, 1974.

[45] Evgeny S Levitin and Boris T Polyak. Constrained minimization methods. *USSR Computational mathematics and mathematical physics*, 6(5):1–50, 1966.

[46] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.

[47] Euhanna Ghadimi, Hamid Reza Feyzmahdavian, and Mikael Johansson. Global convergence of the heavy-ball method for convex optimization. In *Proc. of European control conference*, pages 310–315, 2015.

[48] Y Nesterov. A method of solving a convex programming problem with convergence rate $1/k^2$. In *Soviet Math. Dokl*, volume 27, 1983.

[49] Guanghui Lan and Yi Zhou. Conditional gradient sliding for convex optimization. *SIAM Journal on Optimization*, 26(2):1379–1409, 2016.

[50] Alexander Schwing, Tamir Hazan, Marc Pollefeys, and Raquel Urtasun. Globally convergent parallel MAP LP relaxation solver using the Frank-Wolfe algorithm. In *Proc. Intl. Conf. on Machine Learning*, pages 487–495, 2014.

[51] Kenneth L Clarkson. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4):63, 2010.

[52] Yu Nesterov. Complexity bounds for primal-dual methods minimizing the model of objective function. *Mathematical Programming*, 171(1-2):311–330, 2018.

[53] Jelena Diakonikolas and Lorenzo Orecchia. The approximate duality gap technique: A unified theory of first-order methods. *SIAM Journal on Optimization*, 29(1):660–689, 2019.

[54] Brendan O'donoghue and Emmanuel Candes. Adaptive restart for accelerated gradient schemes. *Foundations of computational mathematics*, 15(3):715–732, 2015.

[55] Dan Garber and Elad Hazan. Faster rates for the Frank-Wolfe method over strongly-convex sets. In *Proc. Intl. Conf. on Machine Learning*, 2015.

[56] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *arXiv preprint arXiv:1804.09554*, 2018.

[57] Mingrui Zhang, Zebang Shen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. One sample stochastic Frank-Wolfe. In *Proc. Intl. Conf. on Artificial Intelligence and Statistics*, pages 4012–4023. PMLR, 2020.

[58] Bingcong Li, Mario Coutino, Georgios B Giannakis, and Geert Leus. A momentum-guided Frank-Wolfe algorithm. *IEEE Trans. on Signal Processing*, 69:3597–3611, 2021.

[59] Jacob D Abernethy and Jun-Kun Wang. On Frank-Wolfe and equilibrium computation. In *Proc. Advances in Neural Info. Process. Syst.*, pages 6584–6593, 2017.

[60] Bingcong Li, Lingda Wang, Georgios B Giannakis, and Zhizhen Zhao. Enhancing Frank Wolfe with an extra subproblem. In *Proc. of AAAI Conf. on Artificial Intelligence*, 2021.

[61] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.

[62] Bo Liu, Xiao-Tong Yuan, Shaoting Zhang, Qingshan Liu, and Dimitris N Metaxas. Efficient k-support-norm regularized minimization via fully corrective Frank-Wolfe method. In *Proc. Intl. Joint Conf. on Artifical Intelligence*, pages 1760–1766, 2016.

[63] Yilang Zhang, Bingcong Li, and Georgios B Giannakis. Accelerating frank-wolfe with weighted average gradients. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5529–5533. IEEE, 2021.

[64] Bingcong Li, Mario Coutiño, and Georgios B Giannakis. Revisit of estimate sequence for accelerated gradient methods. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3602–3606. IEEE, 2020.

[65] Fabian Pedregosa, Armin Askari, Geoffrey Negiar, and Martin Jaggi. Step-size adaptivity in projection-free optimization. *arXiv preprint arXiv:1806.05123*, 2018.

[66] Gábor Braun, Sebastian Pokutta, Dan Tu, and Stephen Wright. Blended conditional gradients: the unconditioning of conditional gradients. *arXiv preprint arXiv:1805.07311*, 2018.

[67] Dan Garber and Ofer Meshi. Linear-memory and decomposition-invariant linearly convergent conditional gradient algorithm for structured polytopes. In *Proc. Advances in Neural Info. Process. Syst.*, pages 1001–1009, 2016.

[68] Thomas Kerdreux, Alexandre d'Aspremont, and Sebastian Pokutta. Projection-free optimization on uniformly convex sets. *arXiv preprint arXiv:2004.11053*, 2020.

[69] Jacob Abernethy, Kevin A Lai, Kfir Y Levy, and Jun-Kun Wang. Faster rates for convex-concave games. In *Conference On Learning Theory*, pages 1595–1625, 2018.

[70] Francis Bach. On the effectiveness of richardson extrapolation in machine learning. *arXiv preprint arXiv:2002.02835*, 2020.

[71] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[72] Yu Nesterov. Universal gradient methods for convex optimization problems. *Mathematical Programming*, 152(1-2):381–404, 2015.

[73] Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. *arXiv preprint arXiv:1407.1537*, 2014.

[74] Walid Krichene, Alexandre Bayen, and Peter L Bartlett. Accelerated mirror descent in continuous and discrete time. In *Proc. Advances in Neural Info. Process. Syst.*, pages 2845–2853, 2015.

[75] Atsushi Nitanda. Stochastic proximal gradient descent with acceleration techniques. In *Proc. Advances in Neural Info. Process. Syst.*, pages 1574–1582, Montreal, Canada, 2014.

[76] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Proc. Advances in Neural Info. Process. Syst.*, pages 3384–3392, Montreal, Canada, 2015.

[77] Weijie Su, Stephen Boyd, and Emmanuel Candes. A differential equation for modeling Nesterov accelerated gradient method: Theory and insights. In *Proc. Advances in Neural Info. Process. Syst.*, pages 2510–2518, 2014.

[78] Jingzhao Zhang, Aryan Mokhtari, Suvrit Sra, and Ali Jadbabaie. Direct runge-kutta discretization achieves acceleration. In *Proc. Advances in Neural Info. Process. Syst.*, pages 3900–3909, 2018.

[79] Bin Shi, Simon S Du, Weijie J Su, and Michael I Jordan. Acceleration via symplectic discretization of high-resolution differential equations. *arXiv preprint arXiv:1902.03694*, 2019.

[80] Andrei Kulunchakov and Julien Mairal. Estimate sequences for variance-reduced stochastic composite optimization. In *Proc. Intl. Conf. on Machine Learning*, 2019.

[81] Bingcong Li, Lingda Wang, and Georgios B Giannakis. Almost tune-free variance reduction. In *Proc. Intl. Conf. on Machine Learning*, 2020.

[82] Zhaoyue Chen, Mokhwa Lee, and Yifan Sun. Continuous time frank-wolfe does not zig-zag. *arXiv preprint arXiv:2106.05753*, 2021.

[83] Arkadi 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*, 15(1):229–251, 2004.

[84] Joseph C Dunn. Rates of convergence for conditional gradient algorithms near singular and nonsingular extremals. *SIAM Journal on Control and Optimization*, 17(2):187–211, 1979.

[85] Lijun Ding, Yingjie Fei, Qiantong Xu, and Chengrun Yang. Spectral Frank-Wolfe algorithm: Strict complementarity and linear convergence. In *Proc. Intl. Conf. on Machine Learning*, 2020.

[86] Dan Garber. Revisiting Frank-Wolfe for polytopes: Strict complementary and sparsity. *arXiv preprint arXiv:2006.00558*, 2020.

[87] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proc. Intl. Conf. on Machine Learning*, pages 272–279. ACM, 2008.

[88] GM Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon: translations of Russian and East European mathematical economics*, 12:747–756, 1976.

[89] Jelena Diakonikolas and Lorenzo Orecchia. Accelerated extra-gradient descent: A novel accelerated first-order method. *arXiv preprint arXiv:1706.04680*, 2017.

[90] Ali Kavis, Kfir Y Levy, Francis Bach, and Volkan Cevher. Unixgrad: A universal, adaptive algorithm with optimal guarantees for constrained optimization. In *Proc. Advances in Neural Info. Process. Syst.*, pages 6257–6266, 2019.

[91] Gábor Braun, Sebastian Pokutta, and Daniel Zink. Lazifying conditional gradient algorithms. In *Proc. Intl. Conf. on Machine Learning*, pages 566–575. PMLR, 2017.

[92] Gábor Braun, Sebastian Pokutta, Dan Tu, and Stephen Wright. Blended conditonal gradients. In *Proc. Intl. Conf. on Machine Learning*, pages 735–743. PMLR, 2019.

[93] Yura Malitsky and Konstantin Mishchenko. Adaptive gradient descent without descent. *arXiv preprint arXiv:1910.09529*, 2019.

[94] Geoffrey Négiar, Gideon Dresdner, Alicia Tsai, Laurent El Ghaoui, Francesco Locatello, Robert Freund, and Fabian Pedregosa. Stochastic frank-wolfe for constrained finite-sum minimization. In *Proc. Intl. Conf. on Machine Learning*, pages 7253–7262. PMLR, 2020.

[95] Francesco Locatello, Alp Yurtsever, Olivier Fercoq, and Volkan Cevher. Stochastic frank-wolfe for composite convex minimization. *Proc. Advances in Neural Info. Process. Syst.*, 32, 2019.

[96] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Proc. Advances in Neural Info. Process. Syst.*, pages 315–323, Lake Tahoe, Nevada, 2013.

[97] Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

[98] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *Proc. Intl. Conf. Machine Learning*, Sydney, Australia, 2017.

[99] Bingcong Li, Meng Ma, and Georgios B Giannakis. On the convergence of SARAH and beyond. *arXiv preprint arXiv:1906.02351*, 2019.

[100] Alp Yurtsever, Suvrit Sra, and Volkan Cevher. Conditional gradient methods via stochastic path-integrated differential estimator. In *Proc. Intl. Conf. on Machine Learning*, pages 7282–7291. PMLR, 2019.

[101] Elad Hazan and Haipeng Luo. Variance-reduced and projection-free stochastic optimization. In *Proc. Intl. Conf. on Machine Learning*, pages 1263–1271. PMLR, 2016.

[102] Quentin Berthet and Vianney Perchet. Fast rates for bandit optimization with upper-confidence frank-wolfe. *Proc. Advances in Neural Info. Process. Syst.*, 30, 2017.

[103] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[104] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

[105] Bingcong Li, Tianyi Chen, and Georgios B Giannakis. Bandit online learning with unknown delays. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 993–1002. PMLR, 2019.

[106] Lingda Wang, Huozhi Zhou, Bingcong Li, Lav R Varshney, and Zhizhen Zhao. Near-optimal algorithms for piecewise-stationary cascading bandits. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3365–3369. IEEE, 2021.

[107] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P Gummadi. Fairness constraints: A flexible approach for fair classification. *The Journal of Machine Learning Research*, 20(1):2737–2778, 2019.